MASTER'S DEGREE THESIS

Interuniversity Master in Statistics and Operations Research

Title: Analytical and Graphical Goodness of Fit Methods for Parametric Survival Models with Right-censored Data

Author: Anna Febrer Galvany

Advisor: Guadalupe Gómez Melis

Co-advisor: Klaus Gerhard Langohr

Department: Departament d'Estadística i Investigació Operativa

University: Universitat Politècnica de Catalunya - Universitat de Barcelona

Academic year: 2014-2015

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

UNIVERSITAT DE BARCELONA

Universitat Politècnica de Catalunya
Facultat de Matemàtiques i Estadística

Master's degree thesis

Analytical and Graphical Goodness of Fit Methods for Parametric Survival Models with Right-censored Data

Anna Febrer Galvany

Advisor: Guadalupe Gómez Melis

Co-advisor: Klaus Gerhard Langohr

Departament d'Estadística i Investigació Operativa

Contents

Chapter 1
Introduction

In survival analysis, the interest is focused on studying the time to a certain event, often called failure and denoted by $\mathcal{E}$. This time to the failure is known as failure time. Examples of failure times include the lifetime of machine components in industrial reliability, the durations of strikes or periods of unemployment in economics, the time taken by subjects to complete specific tasks in psychological experimentation, survival times of patients in a clinical trial, among others.

The study of the failure time is done via a group of subjects measuring the length time before they fail. But to have a precisely determined failure time, there are three requirements:

- a time origin must be unambiguously defined,
- a scale for measuring the elapsed time must be agreed and
- the meaning of the failure $\mathcal{E}$ must be clearly specified.

With this three conditions, the failure time is completely specified and can be modelled with a non-negative random variable, say T. Each subject can fail at most once and, considering all the subjects, the set of their failure times, denoted by $t_1, t_2, \ldots, t_n$ where n is the number of subjects in the study, is known as time-to-event data.

Time-to-event data often present a peculiar feature known as censoring. The presence of censoring complicates the analysis of such data. Censoring, broadly speaking, occurs when some lifetimes are known to have occurred only within certain intervals. There are various types of censoring, such as right censoring, left censoring, and interval censoring; but in this work we will only consider right censoring.

1.1. Right Censoring

Let us say we have a subject who is observed, failure-free, for three weeks and then withdrawn from the study. We do not know exactly when the subject will fail, so we neither cannot measure the failure time of the subject. But we can state that the subject has a failure time which must exceed three weeks. That is, assuming that weeks is the measuring scale, we know that the event should occur in the interval

$(3, \infty)$. In this case, it is said that the observation of the subject's failure time is right-censored.

Note that, like failure, censoring can be seen as an event and the period of observation for censored individuals must also be recorded. Hence the time to censoring can also be modelled with a non-negative random variable, say C.

In this kind of censoring what we observe is the minimum between the failure time and the censoring time. Moreover, we know if this minimum corresponds to the failure or to the censoring time. Let us suppose that, in the absence of censoring, the ith subject in the sample of n has failure time T_i, where $T_1, \ldots, T_n$ are independent and identically distributed with unknown distribution function F. Let us also suppose that there is a period of observation C_i such that observation of the subject stops at C_i if failure has not occurred by then. Then the observation consists of $Y_i = \min(T_i, C_i)$, together with the indicator of censoring δ_i, pointing if we have observed a failure or a censoring time, defined as

$$\delta_i = \begin{cases} 1 & \text{if } T_i \leq C_i, \\ 0 & \text{if } T_i > C_i. \end{cases}$$

That is, if $\delta_i = 1$ the event is not censored and $Y_i = T_i$ whilst if $\delta_i = 0$ the event is censored and $Y_i = C_i$. These data are conveniently represented by the pairs (Y_i, δ_i), for $i = 1, 2, \ldots, n$.

Within right censoring, one can distinguish four different types of censoring depending on the pattern of the censoring times. These types are the following:

- Fixed Type I censoring: In this type of censoring the censoring time is fixed. That is, there is a preassigned observation time C_R, equal for all the individuals, which enter at the study at the same time. Hence what we observe is (Y_i, δ_i) for i in $1, \ldots, n$ where

$$Y_i = \min(T_i, C_R) \qquad \text{and} \qquad \delta_i = \begin{cases} 1 & \text{if } T_i \leq C_R, \\ 0 & \text{if } T_i > C_R. \end{cases}$$

  In fixed Type I censoring, the number of observed events is random since we do not know how many failures will occur during the study.

- Generalized Type I censoring: In some studies not all the individuals enter the study at the same moment. This type of censoring appears, for example, when the end of the study is established at C_R and each individual enters the study at different time $\mathcal{O}_i$. The potential time to failure F_i of each individual will only be observed if $\mathcal{E}$ occurs before C_R. To analyse the data, it is interesting to consider the entry time of each individual as 0, hence a rescaling of the variables is needed. Let us define $T_i = F_i - \mathcal{O}_i$ and $C_i = C_R - \mathcal{O}_i$. So, although we have a fixed period of observation C_R, each individual has their own censoring time C_i that can differ from one individual to another. Then

we observe (Y_i, δ_i) for i in $1, \ldots, n$ where

$$Y_i = \min(T_i, C_i) \qquad \text{and} \qquad \delta_i = \begin{cases} 1 & \text{if } T_i \leq C_i, \\ 0 & \text{if } T_i > C_i. \end{cases}$$

As in fixed Type I censoring, the number of observed events in this type of censoring is random and remains unknown until the end of the study.

- TYPE II CENSORING: In this type of censoring the end of the study is not preset. The study ends after a certain number of failures has occurred, say r ($r < n$). That is, the observation period goes from the beginning until the time of the rth failure.

  Unlike fixed and generalized Type I censoring, in Type II censoring the number of observed events is not random but fixed. In this case, what is random is the censoring time C that takes the value of the rth failure time $T_{(r)}$. So, in this type of censoring, the ordered observed pairs are $(Y_{(i)}, \delta_{(i)})$ where $Y_{(i)} = \min(T_{(i)}, T_{(r)})$ and $\delta_{(i)}$ is defined as

$$\delta_{(i)} = \begin{cases} 1 & \text{if } T_{(i)} \leq T_{(r)}, \\ 0 & \text{if } T_{(i)} > T_{(r)}. \end{cases}$$

- RANDOM CENSORING: In random censoring, both failure and censoring times are treat as random variables following unknown distributions. Then, as we said before, the $T_1, \ldots, T_n$ failure times are independent and identically distributed with unknown distribution function F, but also the $C_1, \ldots, C_n$ censoring times are independent and identically distributed with unknown distribution function G. In random censoring we observe the pairs (Y_i, δ_i) for i in $1, \ldots, n$ where $Y_i = \min(T_i, C_i)$, being $C_1, \ldots, C_n$ the censoring times for each individual and

$$\delta_i = \begin{cases} 1 & \text{if } T_i \leq C_i, \\ 0 & \text{if } T_i > C_i. \end{cases}$$

  In order to assume random censoring, the independence between T_i and C_i is needed. Typical examples of where the random censoring times may be thought to be independent of the main event time of interest are accidental deaths, migration of human population, ans so forth.
  Throughout this work we will assume that T and C are independent and inference will be done under this assumption.

In many medical studies, the censoring scheme is a combination of random and Type I censoring. In such studies, some patients are randomly censored when, for example, they move from the study location for reasons unrelated to the event of interest, whereas others are Type I censored when the fixed study period ends.

On the other hand, Type II censoring is most often used in testing of equipment life. Here, all items are put on test at the same time, and the test is terminated when r of the n items have failed. Such an experiment may save time and money because it could take very long time for all items to fail.

1.2. Survival and Hazard functions

The main function employed to describe time-to-event phenomena is the *survival function*, $S(t)$, which is the probability of an individual to survive longer than time t. It is defined as

$$S(t) = P(T > t), \quad t \geq 0.$$

Note that, since the distribution function is $F(t) = P(T \leq t)$, the survival function can be expressed as

$$S(t) = 1 - F(t) = \sum_{t_j > t} P[T = t_j]$$

when the time T is discrete and takes the values $t_1 < t_2 < \ldots$ and

$$S(t) = 1 - F(t) = 1 - \int_0^t f(u)du = \int_t^\infty f(u)du$$

when the time T is continuous and $f(t)$ is the density function.

Survival curves can have many different forms but all have the same properties. They are monotone and decreasing functions and satisfy $S(0) = 1$ and $\lim_{t\to\infty} S(t) = 0$. Their rate of decline varies according to the risk of experiencing the event at time t but it is difficult to determine the essence of a failure pattern by simply looking at the survival curve. Nevertheless, this function continues to be a very popular description of survival in the applied literature and can be very useful in comparing two or more mortality patterns.

Another basic quantity, fundamental in survival analysis and whose curve is indeed very useful to get an idea of the essence of the failure pattern, is the *hazard function* $\lambda(t)$. This function is also known as the conditional failure rate in reliability, the force of mortality in demography, the intensity function in stochastic processes, the age-specific failure rate in epidemiology, the inverse of the Mill's ratio in economics, or simply as the hazard rate. The hazard function which computes the risk to fail at time t assuming that the individual has not failed before. It can be expressed as

$$\lambda(t_j) = P[T = t_j \mid T \geq t_j] = P[T = t_j \mid T > t_{j-1}] \tag{1.1}$$

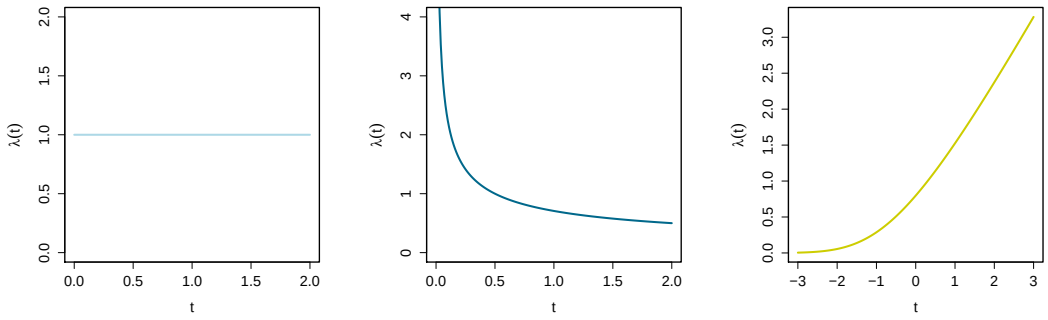
when the time T is discrete and takes the values $t_1 < t_2 < \ldots$ and

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{P[t \leq T < t + \Delta t \mid T \geq t]}{\Delta t} \tag{1.2}$$

when the time T is continuous. Note that the hazard function is a probability in the discrete case but not in the continuous case. So when T is discrete $\lambda(t)$ is bounded by 0 and 1, but when T is continuous these boundaries are no longer applicable.

From (1.1) and (1.2), one can see that $\lambda(t)$ for the discrete case and $\lambda(t)\Delta t$ for the continuous case may be viewed as the probability ("approximate" in the continuous case) of an individual that has not failed by t to experience the event in the next instant. This function is particularly useful in determining the appropriate failure distribution utilizing qualitative information about the mechanism of failure and for describing the way in which the chance of experiencing the event changes with time. There are many general shapes for the hazard function. The only restriction on $\lambda(t)$ is that it must be nonnegative, i.e., $\lambda(t) \geq 0$.

Some generic types of hazard functions are plotted in Figure 1.1. For example, one may believe that the hazard rate for the occurrence of a particular event is constant, decreasing, increasing, hump-shaped, bathtub-shaped, or possessing some other characteristic that describes the failure mechanism.



(A) Constant hazard (B) Decreasing hazard (C) Increasing hazard

(D) Hump shaped hazard (E) Bathtub shaped hazard

FIG. 1.1. Five possible shapes for the hazard function

Models with increasing hazard rates may arise when there is natural ageing or wear. Decreasing hazard functions are much less common but find occasional use when there is a very early likelihood of failure, such as in certain types of electronic devices or in patients experiencing certain types of transplants. Most often, a bathtub-shaped hazard is appropriate in populations followed from birth. Similarly, some manufactured equipment may experience early failure due to faulty parts, followed by a constant hazard rate which, in the later stages of equipment life, increases. Most population mortality data follow this type of hazard function where, during an early period, deaths result, primarily, from infant diseases, after which the death rate stabilises, followed by an increasing hazard rate due to the natural ageing process. Finally, if the hazard rate is increasing early and eventually begins declining, then, the hazard is termed hump-shaped. This type of hazard rate is often used in modelling survival after successful surgery where there is an initial increase in risk due to infection, hemorrhaging, or other complications just after the procedure, followed by a steady decline in risk as the patient recovers. Specific

distributions that give rise to these different types of failure rate are presented in Chapter 2.

Another measure of risk is the *cumulative hazard function* $\Lambda(t)$, which is defined as

$$\Lambda(t) = \sum_{t_j \leq t} \lambda(t_j)$$

when the time T is discrete and takes the values $t_1 < t_2 < \ldots$ and

$$\Lambda(t) = \int_0^t \lambda(u)du$$

when the time T is continuous.

This measure can be interpreted as the greater the value of $\Lambda(t)$, the greater the risk of failure by time t. The cumulative hazard function is a non-negative function and monotonically increasing. Using a nonparametric estimator of $\Lambda(t)$, developed in Section 1.3, The relationship between some transformation of the cumulative hazard function and some function of time has been exploited to provide a graphical check of the goodness of fit of a distribution to data. This idea is explained more detailed in Section 3.2.

We have seen that the hazard function and the cumulative hazard function are related, but there also exists a relation between the cumulative hazard function and the survival function. In the continuous case, these two function can be related in the following way,

$$\Lambda(t) = -\ln S(t),$$

or equivalently,

$$S(t) = e^{-\Lambda(t)}.$$

Note that for the discrete case these two equalities do not hold. It is for this reason that some authors such as Cox and Oakes, 1984 [**CO84**] redefine the cumulative hazard function as

$$\Lambda(t) = -\sum_{t_j \leq t} \log\left[1 - \lambda(t_j)\right]$$

to ensure that $S(t) = e^{-\Lambda(t)}$.

The relations between the hazard and the survival function are the following. In the continuous case we have that

$$S(t) = e^{-\int_0^t \lambda(u)du}$$

and

$$f(t) = \lambda(t)S(t).$$

And in the discrete case the hazard and the survival can be related by

$$S(t_j) = S(t_{j-1})(1 - \lambda(t_j)), \quad j = 1, 2, \ldots$$

taking $t_0 = 0$.

1.3. Estimation of the Survival and the Cumulative Hazard function

If we are dealing with complete data sets, the survival function can easily be estimated by

$$\hat{S}(t) = 1 - \hat{F}(t), \tag{1.3}$$

where $\hat{F}$ is the empirical distribution function. Unfortunately, if we have censored observations in the sample, the empirical distribution function is no longer a consistent estimator of the theoretical distribution function, so the estimation introduced above can not be applied. Hence in this situation other ways to estimate the survival function must be used.

Before explaining how we can estimate the survival and the cumulative hazard function for right-censored samples, the introduction of some notation is needed. Note that in order to adjust most to reality, we will take in account that these can be ties with the observations. Namely that we can have more than one individual failing at the same time or that censored and non-censored observations occur at the same time. In this last case we will assume that the non-censored observations take place just before the censored ones.

$Y_{(1)} < \ldots < Y_{(i)} < \ldots < Y_{(r)}$: the r different times,

n_i : number of individuals that are at risk just before $Y_{(i)}$,

d_i : number of individuals that fail at moment $Y_{(i)}$.

The product-limit estimator of the survival function introduced by Kaplan and Meier (1985) [**KM55**] is perhaps the most commonly used estimator for censored data. This estimator is known as *Kaplan-Meier estimator* ($\hat{S}_{KM}$) and is given by:

$$\hat{S}_{KM}(t) = \begin{cases} 1 & \text{if } t < Y_{(1)} \\ \displaystyle\prod_{i:Y_{(i)} \leq t} \left(1 - \frac{d_i}{n_i}\right) & \text{if } t \geq Y_{(1)} \end{cases}$$

Note that the Kaplan-Meier estimator is not well defined when the last observation is censored, since in this case the survival function never takes the value 0 and $\lim_{t\to\infty} \hat{S}_{KM}(\infty) > 0$. In order to deal with it Efron suggested to redefine $\hat{S}_{KM}(t) = 0$ for all $t \geq Y_{(n)}$. Otherwise, Gill suggested to maintain $\hat{S}_{KM}(t) = \hat{S}_{KM}(Y_{(n)})$ when $\delta_{(n)} = 0$ for all $t > Y_{(n)}$. Although both suggestions have the same behaviour asymptotically, the suggestion of Gill has a better behaviour for small samples.

The Kaplan-Meier estimator is a step function with jumps at the observed event times. The size of these jumps depends not only on the number of events observed at each time $Y_{(i)}$, but also on the pattern of the censored observations prior to $Y_{(i)}$.

When the data does not have censored observations, the Kaplan-Meier estimator reduces to the empirical survival function introduced in (1.3).

Since there exists a relation between the survival and the cumulative hazard function, an estimator of $\Lambda(t)$ based on the Kaplan-Meier estimator can be computed. This estimator of the cumulative hazard is given by

$$\hat{\Lambda}_{KM}(t) = -\ln \hat{S}_{KM}(t).$$

Another estimation of the cumulative hazard function, which performs better for small sample sizes than the one based on Kaplan-Meier, is the *Nelson-Aalen estimator*. The estimator was first suggested by Nelson (1972) [**Nel72**] in a reliability context and later rediscovered by Aalen (1978) [**Aal78**], who derived the estimator using modern counting process techniques. It is given by:

$$\hat{\Lambda}_{NA}(t) = \begin{cases} 0 & \text{if } \ t < Y_{(1)} \\ \displaystyle\sum_{i:Y_{(i)} \leq t} \frac{d_i}{n_i} & \text{if } \ t \geq Y_{(1)} \end{cases}$$

In the same way that one can compute an estimator of the cumulative hazard function based on the Kaplan-Meier estimator, one can also define an estimator of the survival function based on the Nelson-Aalen estimator. This estimator is given by

$$\hat{S}_{NA}(t) = e^{-\hat{\Lambda}_{NA}(t)}.$$

The Nelson-Aalen estimator has two primary uses in analysing survival data. The first one is to select between parametric models for the failure time. Here the Nelson-Aalen estimator is transformed in such a way that the graph of plotting this transformation against some function of t will be approximately linear if the given parametric model fits the data. In Section 3.2, this method is explained with more detail.

The second use of the Nelson-Aalen estimator is to provide crude estimates of the hazard rate $\lambda(t)$. These estimates are the slope of the Nelson-Aalen estimator, but better estimates can be obtained by smoothing the jump sizes of the Nelson-Aalen estimator with a parametric kernel.

When multiple death are simultaneous, both Kaplan-Meier and Nelson-Aalen estimators can be modified to treat the simultaneous death times as though they were in fact distinct, even if the distinction is unknown. The Kaplan-Meier estimator when we want to break the ties is defined as

$$\hat{S}_{KM}(t) = \begin{cases} 1 & \text{if } \ t < Y_{(1)} \\ \displaystyle\prod_{i:Y_{(i)} \leq t} \prod_{k=0}^{d_i - 1} \left(1 - \frac{1}{n_i - k}\right) & \text{if } \ t \geq Y_{(1)} \end{cases},$$

and the Nelson-Aalen estimator when we want to break the ties is defined as

$$\hat{\Lambda}_{NA}(t) = \begin{cases} 0 & \text{if } t < Y_{(1)} \\ \displaystyle\sum_{i:Y_{(i)} \leq t} \sum_{k=0}^{d_i - 1} \frac{1}{n_i - k} & \text{if } t \geq Y_{(1)} \end{cases} . \tag{1.4}$$

1.4. Parametric Survival Models

In this work we will focus on parametric survival models, that is we are willing to assume a parametric form for the distribution of the survival time. Using parametric models instead of non-parametric ones has its advantages and disadvantages. The advantages are that the estimation of $S(t)$ is easier and estimated survival curves are smoother than nonparametric estimates. However, the main disadvantage of parametric methods is that they require extra assumptions that may not be appropriate, and the choice of an inappropriate model can lead to incorrect results.

In parametric models we are assuming that the data com from a certain distribution $F_0(\,\cdot\,, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the vector of the distribution's parameters ($\boldsymbol{\theta} \in \Omega$). Depending on the knowledge that we have about the data, the parameters of F_0 can be known – or intuited – or can be completely unknown. In this last case, in order to facilitate the statistic inference, the unknown parameters are usually replaced by an estimator based on the observed data.

The most common estimator of the parameters is the *maximum likelihood estimator*, which is the one that maximise the likelihood function.

The *likelihood function* is represented as a product of the contribution of each individual. Let $T_1, \ldots, T_n$ be independent and identically distributed random variables with survival function $S(\,\cdot\,, \boldsymbol{\theta})$ and density function $f(\,\cdot\,, \boldsymbol{\theta})$ with $\boldsymbol{\theta} \in \Omega$; and let $C_1, \ldots, C_n$ be iid random variables with survival function $G(\,\cdot\,, \boldsymbol{\varphi})$ and density function $g(\,\cdot\,, \boldsymbol{\varphi})$ with $\boldsymbol{\varphi} \in \Phi$. T_i are the event times while C_i are the censoring times. Finally let $Y_i = \min(T_i, C_i)$ be the observed times and $\delta_i = 1(T_i \leq C_i)$ the censoring indicator. Then the likelihood function for right-censored data is

$$\mathcal{L}_i(\boldsymbol{\theta}, \boldsymbol{\varphi} | \boldsymbol{y}) = \prod_{i=1}^{n} f(y_i|\boldsymbol{\theta})^{\delta} S(y_i|\boldsymbol{\theta})^{1-\delta_i} \prod_{i=1}^{n} G(y_i|\boldsymbol{\varphi})^{\delta} g(y_i|\boldsymbol{\varphi})^{1-\delta_i}$$

When $S(\,\cdot\,, \boldsymbol{\theta})$ and $G(\,\cdot\,, \boldsymbol{\varphi})$ are functionally independent, the two products can be maximised independently. Hence, in this case, inference can be simply based on

$$\mathcal{L}(\boldsymbol{\theta} | \boldsymbol{y}) = \prod_{i=1}^{n} f(y_i)^{\delta_i} S(y_i)^{1-\delta_i}$$

From the likelihood function one can get the *maximum likelihood estimator* $\hat{\boldsymbol{\theta}}_{MLE}$, which is the random variable that maximises the likelihood function. That is, $\hat{\boldsymbol{\theta}}_{MLE}$ meets

$$\left. \frac{\partial \mathcal{L}(\boldsymbol{\theta} | \boldsymbol{y})}{\partial \theta_i} \right|_{\hat{\boldsymbol{\theta}}_{MLE}} = 0, \qquad \text{for } i = 1, \ldots, r,$$

where $r = \dim(\Omega)$ and $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_r)$.

1.5. State of the art: Goodness of Fit Tests for Right-censored Data

When one is dealing with parametric models it is needed to verify in some way if the chosen model fits well the data set. This verification is done by assessing the goodness of fit of the model, and it can be done graphically or analytically.

As graphical methods to assess the goodness of fit of a model we can find the cumulative hazard plots and the probability plots P-P and Q-Q plots. Another probability plot called stabilised probability plot was introduced by Micheal (1983) [**Mic83**] as a transformation of a P-P plot to stabilise the variance of the plotted points. The graphical methods presented above where developed for non-censored data but they are still valid for right-censored data. However P-P plots, as well as Q-Q plots and stabilised probability plots, present the handicap that when the data have censored observations, the plotted points are not evenly spread. To overcome this problem, Waller and Turnbull (1992) [**WT92**] proposed the empirically rescaled plot.

The goodness of fit can also be assessed analytically via a goodness of fit statistic. Most goodness of fit statistics can be regarded as measures of proximity between two distributions: the empirical and the hypothesised. For instance, the Kolmogorov-Smirnov statistic introduced by Kolmogorov in 1933 [**Kol33**] is based on the supremum distance, whereas the Cramér-von Mises statistic, proposed by Cramér [**Cra28**] and von Mises [**vM28**] in 1928, and the Anderson-Darling statistics, introduced by Anderson and Darling in 1952 [**AD52**], use a weighted-L^2 distance. These statistics are useful to analyse non-censored data, but for right-censored data their asymptotic distribution is not known. Although workers in survival analysis and reliability theory often have been concerned about their parametric assumptions, relatively few general goodness of fit procedures seem to have been available for time-continuous data when censoring is present.

Dufour and Maag (1978) [**DM78**] and Barr and Davidson (1973) [**BD73**] proposed a modification of the Kolmogorov-Smirnov statistic to test type I censored data. The asymptotic distribution of this modified statistic has been obtained and tabulated by Koziol and Byar (1975) [**KB75**]. Schey (1977) [**Sch81**] also proposed a modification of the one-sided Kolmogorov-Smirnov for type I censoring. Fleming, O'Fallon, O'Brien and Harrington (1980) [**FOOH80**] modified the Kolmogorov-Smirnov statistic but for use with arbitrarily right-censored data.

Pettitt and Stephens (1976) [**PS76**] modified Cramér-von Mises type statistics so that tests of goodness of fit could be made for the simple hypothesis with randomly censored data. The asymptotic theory for the Pettitt and Stephens statistic was studied and developed by Koziol and Green (1976) [**KG76**] when tests of fit are made with unknown parameters.

Chi-square tests for random censoring were developed by Habib and Thomas (1986) [**HT86**] and Kim (1993) [**Kim93**]. Mihalko and Moore (1980) [**MM80**] also proposed a Chi-square test but only suitable for type II censored data.

Hjort (1990) [**Hjo90**] proposed goodness-of-fit tests based on a weighted version of the cumulative hazard process. Turnbull and Weiss (1978) [**TW78**] considered a likelihood ratio statistic applicable for discrete or grouped censored data with finite support. Grané (2012) [**Gra12**] built an statistics based on Hoeffding's maximum correlation for testing type I and type II censored data.

In 1986, D'Agostino and Stephens edited the book "Goodness-of-Fit Techniques" [**DS86**] compiled from the leading methods of testing fit studied until then. The book shows how to apply the techniques, emphasises testing for the three major distributions, normal, exponential and uniform, provide tables to make the tests available and discusses the handling of censored data.

1.6. Outline of this Master's Degree Thesis

When one is dealing with parametric models, it is important to choose an appropriate distribution, otherwise the results can be incorrect. So it is important to assess the goodness of fit of the chosen distribution before continue with the analysis. In survival it is very common to have right-censored data, and we found that there are few references about goodness of fit methods for such data. It is for this reason that this master's degree thesis is a compilation of some methods to assess goodness of fit when data is right-censored. We chose the methods we find more interesting, we studied them and we explained how they work. We also implement the methods in `R` with the aim to create a local library for testing goodness of fit for right-censored data.

In Chapter 2 we present the parametric models considered in this work. Among these models one can find well-known and widely used distributions such as the Weibull, the Log-normal and the Log-logistic distributions. Nevertheless, other distributions less known are also introduced since they accept interesting shapes in their hazard functions.

In Chapter 3 four methods for assessing goodness of fit are introduced. Of these four methods two are graphical and two are analytical. The graphical methods presented are the ones found in the literature, the cumulative hazard plots and the probability plots (including the stabilising probability plot and the empirically rescaled plot). The analytical methods included in the chapter are the exact goodness of fit test for type I and type II censored data proposed in Grané (2012) [**Gra12**] and the modified Kolmogorov-Smirnov test for randomly censored data introduced by Fleming et al. (1980) [**FOOH80**]. We decided to focus in these two goodness of fit tests for the following reasons. The Grané's article is by far the most recent contribution in this field and it provided an algorithm for computing the distribution of the statistic that can be implemented to `R`. On the other hand, since the Kolmogorov-Smirnov test is perhaps the most used goodness of fit test, we found interesting to present a modification of this statistic that can be used to test randomly right-censored data.

Finally in Chapter 4 there are explained the use and the details of the `R` implementation of the methods presented in Chapter 3.

Chapter 2
Common parametric models for survival data

In this chapter we are going to introduce some of the most used distributions in survival analysis. These distributions are commonly chosen by investigators for its simplicity in the formulae or because of their flexibility to fit a wide range of cases. Some of the important models discussed are the Weibull, normal, log-normal, log-logistic and the exponential power distributions. The hazard rate is a very important feature in survival, and it is for this reason that we have included for each of the five considered behaviours of the hazard function (constant, increasing, decreasing, hump-shaped and bathtub-shaped) at least one distribution whose hazard can present this shape.

It has been decided to use a unification of the parameters, so the shape parameters will be denoted as α and γ, the location parameter as μ and the scale parameter as β.

2.1. Weibull distribution

The *Weibull distribution* [Wei(α, β)] (described in detail by Waloddi Weibull in 1951 [**Wei51**]) is a very flexible model for lifetime data capable to accommodate increasing, decreasing or constant hazard rates. This fact, coupled with the model's relatively simple survival, hazard, and probability density functions, have made it a very popular parametric model.

Its density function is given by

$$f(t) = \alpha\beta^{\alpha}t^{\alpha-1}e^{-(\beta t)^{\alpha}}$$

and its survival function is of the form

$$S(t) = e^{-(\beta t)^{\alpha}},$$

where $t \in [0, \infty)$, α denotes the shape parameter and β the scale parameter. Both parameters, α and β, must be positive; i.e. $\alpha, \beta \in (0, +\infty)$.

For this distribution, the cumulative hazard and the hazard function are

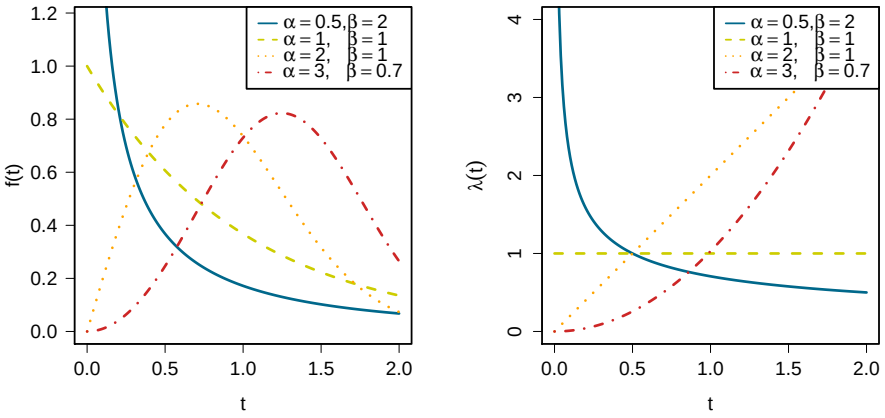
$$\Lambda(t) = (\beta t)^{\alpha}$$

and

$$\lambda(t) = \alpha\beta^{\alpha}t^{\alpha-1}$$

respectively.

The parameter α allows great flexibility of the model and different shapes of the hazard function, which is either increasing, decreasing, or constant depending on the value of α (see some examples in Figure 2.1B). When $\alpha > 1$ the hazard function increases with time, when $\alpha = 1$ the hazard function is constant and when $\alpha < 1$ the hazard function is decreasing over time.



(A) Density functions (B) Hazard functions

FIG. 2.1. Density and hazard functions of the Weibull distribution

Exponential distribution

The *Exponential distribution* [Exp(β)] is a particular case of the Weibull distribution when the shape parameter is 1 ($\alpha = 1$). Its density function is given by

$$f(t) = \beta e^{\beta t}$$

and its survival function by

$$S(t) = e^{-\beta t}.$$

For the exponential distribution, the cumulative hazard and the hazard function are expressed, respectively, as

$$\Lambda(t) = \beta t \qquad \text{and} \qquad \lambda(t) = \beta.$$

The exponential distribution is the simplest parametric model and assumes a constant risk over time, which reflects the property of the distribution appropriately called "lack of memory", also known as "no-ageing" property or "old as good as new" property. Although the exponential distribution has been historically very popular, its constant hazard rate appears too restrictive in both health and industrial applications. An example of the hazard function for the exponential distribution can be seen in Figure 2.1B represented by the green dashed line.

2.2. Gumbel distribution

The *Gumbel distribution* [Gum(μ, β)] is an extreme value distribution, thus it has been applied to many extreme value data such as flood flows, wind speeds, radioactive emissions, etc. The Gumbel distribution is also known as the *Log-weibull distribution* because if T follows a Weibull distribution with shape parameter α and scale parameter β, then $\log T$ follows a Gumbel distribution with location parameter $\mu = -\log \beta$ and scale parameter $\beta = 1/\alpha$.

The survival function for the Gumbel distribution is given by

$$S(t) = e^{-e^{\frac{t-\mu}{\beta}}}$$

where $t \in \mathbb{R}$, μ denotes the location parameter ($\mu \in \mathbb{R}$) and β the scale parameter ($\beta \in (0, \infty)$). Note that in this case, since the domain is all the real line, $S(0) < 1$ and the function takes the value 1 at $t = -\infty$. The density function for this distribution is of the form

$$f(t) = \frac{1}{\beta} e^{\frac{t-\mu}{\beta}} e^{-e^{\frac{t-\mu}{\beta}}}.$$

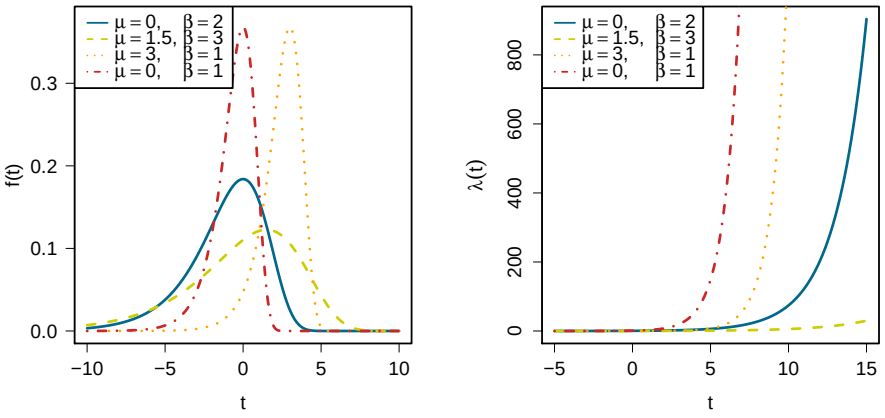
The cumulative hazard function may be written as

$$\Lambda(t) = e^{\frac{t-\mu}{\beta}}$$

and the hazard function as

$$\lambda(t) = \frac{1}{\beta} e^{\frac{t-\mu}{\beta}}.$$

Since the Gumbel distribution does not have any shape parameter, the hazard function always has the same shape whatever are the values of μ and β. This function is always increasing as it can be seen in Figure 2.2B.



(A) Density functions (B) Hazard functions

FIG. 2.2. Density and hazard functions of the Gumbel distribution

2.3. Normal distribution

The *Normal distribution* [Norm(μ, β)] is the most important distribution in statistics. The density function of the Normal distribution is given by

$$f(t) = \frac{1}{\beta\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\beta^2}},$$

and its survival function is defined as

$$S(t) = \int_t^\infty \frac{1}{\beta\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\beta^2}} dx,$$

where $t \in \mathbb{R}$, $\mu \in \mathbb{R}$ is the location parameter, coinciding with the mean of the distribution, and β^2 is the squared scale parameter, coinciding with the variance. Note that this integral does not exist in a simple closed formula and needs to be computed numerically. For this reason and because survival data are often not symmetric, the Normal distribution is not very suitable for this kind of data. For the Normal distribution we also have $S(0) < 1$ and $S(-\infty) = 1$.

The corresponding cumulative hazard function and hazard function are

$$\Lambda(t) = -\log\left(1 - \phi\left(\tfrac{t-\mu}{\beta}\right)\right)$$

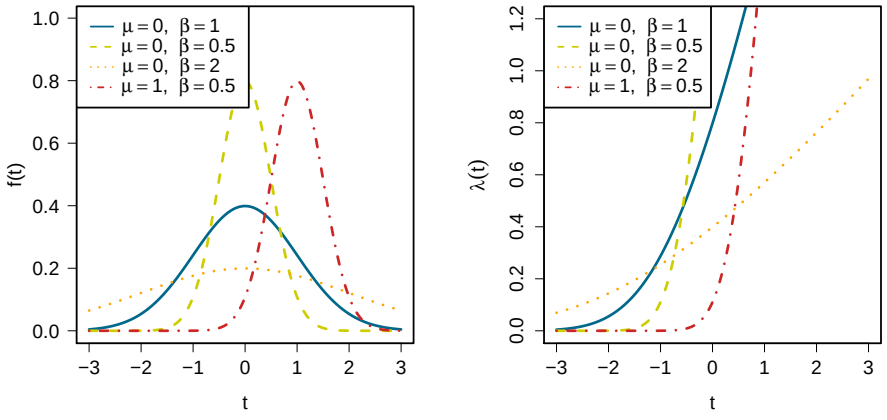
and

$$\lambda(t) = \frac{1}{\beta\sqrt{2\pi}} \cdot \frac{e^{-\frac{(t-\mu)^2}{2\beta^2}}}{1 - \phi(\frac{t-\mu}{\beta})},$$

where ϕ is the so-called *standard normal distribution* (case when $\mu = 0$ and $\beta = 1$) given by

$$\phi(t) = \int_{-\infty}^t \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx.$$

The hazard function for the Normal distribution has always the same shape, independently of the values of the parameters μ and β. Figure 2.3B shows some examples of the hazard function and it can be seen that each of them is increasing.



(A) Density functions (B) Hazard functions

FIG. 2.3. Density and hazard functions of the Normal distribution

2.4. Log-normal distribution

The *Log-normal distribution* [LNorm(μ, β)] is the continuous probability distribution of a random variable whose logarithm follows a normal distribution. That is, if T follows a Log-normal distribution with location parameter μ and scale parameter β then $Y = \log T$ will be normally distributed $Y \sim N(\mu, \tau)$. A Log-normal distribution results from the product of a large number of independent and identically distributed variables in the same way that a normal distribution results from the sum of a large number of independent and identically distributed variables.

The density function of the Log-normal distribution is given by

$$f(t) = \frac{1}{\beta t\sqrt{2\pi}} \exp\left(-\frac{\left[\log t - \mu\right]^2}{2\beta^2}\right).$$

where $t \in (0, \infty)$, $\mu \in \mathbb{R}$ and $\beta > 0$. For this distribution the survival function may be written as

$$S(t) = 1 - \phi\left(\frac{\log t - \mu}{\beta}\right) = \int_{\frac{\log t-\mu}{\beta}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}\, dx,$$

where ϕ is the standard normal distribution introduced before.

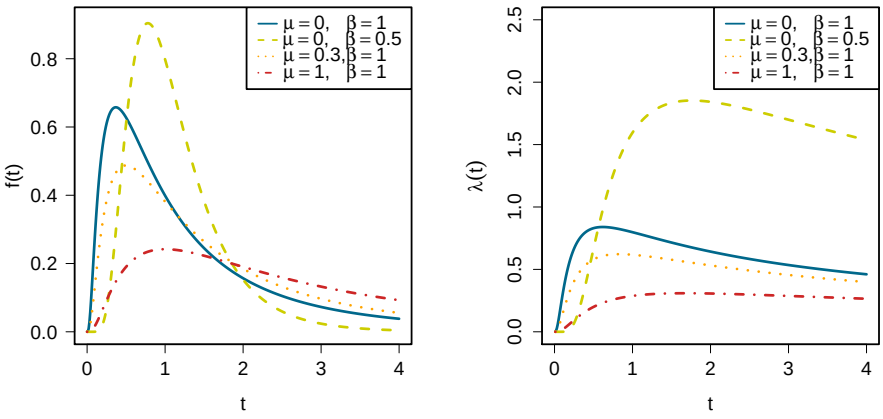
Note that like the Normal distribution, the survival function of the Log-normal is defined as an integral that does not admit a simple closed formula. Hence the Log-normal distribution may be convenient to use with non-censored data, but when this distribution is applied to censored data, where the censored individuals contribute to the likelihood with the survival, the computations quickly become cumbersome.

The cumulative hazard and the hazard function are, respectively, the following:

$$\Lambda(t) = -\log\left[1 - \phi\left(\frac{\log t - \mu}{\beta}\right)\right],$$

$$\lambda(t) = \frac{\frac{1}{\beta t\sqrt{2\pi}} e^{-\frac{(\log t-\mu)^2}{2\beta^2}}}{1 - \phi\left(\frac{\log t-\mu}{\beta}\right)}.$$

The hazard function for the Log-normal is hump-shaped: it has value zero at $t = 0$, increases to a maximum and then decreases, approaching zero as t heads to infinity (see Figure 2.4B). Because of the decreasing form of the hazard function for older ages, the distributions seem implausible as a lifetime model in most situations. Nevertheless, it makes sense if interest is focused on time periods of younger ages. Despite its unattractive features, the Log-normal distribution has been widely used as failure distribution in diverse situations, such as the analysis of electrical insulation or time occurrence of lung cancer among smokers.



(A) Density functions (B) Hazard functions

FIG. 2.4. Density and hazard functions of the Log-normal distribution

2.5. Logistic distribution

The *Logistic distribution* [Logis(μ, β)] is a continuous probability distribution with scale parameter β and location parameter μ. Its cumulative distribution function is the logistic function

$$F(t) = \frac{1}{1 + e^{-\frac{t-\mu}{\beta}}}, \qquad t \in \mathbb{R},\ \mu \in \mathbb{R},\ \beta \in (0, \infty)$$

which appears in logistic regression. The Logistic distribution is closely resembling the normal distribution, but its survival function is mathematically more tractable. This is of the form

$$S(t) = \frac{e^{-\frac{t-\mu}{\beta}}}{1 + e^{-\frac{t-\mu}{\beta}}},$$

and the corresponding density function given by

$$f(t) = \frac{e^{-\frac{t-\mu}{\beta}}}{\beta\left(1 + e^{-\frac{t-\mu}{\beta}}\right)^2}.$$

The shape of $f(t)$ is like the Normal distribution but the Logistic distribution has heavier tails. Like in the Normal distribution, the survival function of the Logistic distribution take a value smaller than 1 at $t = 0$, and the value 1 is reached at $t = -\infty$.

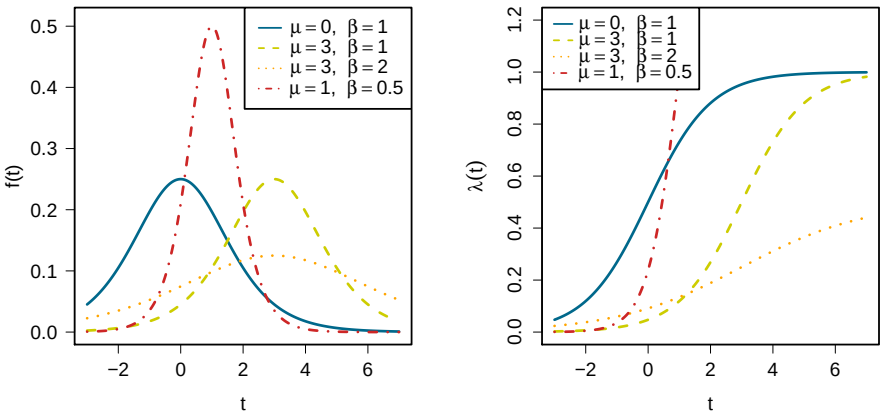
For this distribution, the cumulative hazard function is given by

$$\Lambda(t) = \log\left(1 + e^{\frac{t-\mu}{\beta}}\right),$$

and the hazard function by

$$\lambda(t) = \frac{e^{\frac{t-\mu}{\beta}}}{\beta\left(1 + e^{\frac{t-\mu}{\beta}}\right)}.$$

Like the Normal distribution, the hazard function for the Logistic distribution is always increasing (see Figure 2.5B where some examples of this function are plotted).



(A) Density functions (B) Hazard functions

FIG. 2.5. Density and hazard functions of the Logistic distribution

2.6. Log-logistic distribution

The *Log-logistic distribution* [LLogis(α, β)] is the continuous probability distribution of a random variable whose logarithm follows a logistic distribution. That is if T follows a Log-logistic distribution (α, β) then $Y = \log T$ follows a logistic distribution $(\log \beta, \frac{1}{\alpha})$. The Log-Logistic distribution has a fairly flexible functional form and it is one of the parametric survival time models in which the hazard rate may be decreasing as well as hump-shaped. For this reason this distribution is used in survival analysis as a parametric model for events whose hazard increase initially and decreases later, for example the mortality rate from cancer following diagnosis or treatment.

The survival function for the Log-Logistic distribution is given by

$$S(t) = \frac{1}{1 + \left(\frac{t}{\beta}\right)^{\alpha}},$$

and its density function by

$$f(t) = \frac{\alpha\, t^{\alpha-1} \beta^{-\alpha}}{\left[1 + \left(\frac{t}{\beta}\right)^{\alpha}\right]^2}$$

where $t \in [0, \infty)$, $\beta > 0$ denotes the scale parameter and $\alpha > 0$ the shape parameter.

For $\alpha > 1$ the Log-logistic distribution is very similar in shape to the Log-normal distribution, but it is more suitable for the use in the analysis of survival data. This

is because of its greater mathematical tractability when dealing with the censored observations which occur frequently in such data. Note that the contribution made by a right-censored observation to the likelihood, which is equal to the value of the survival function at the time of censoring, can be evaluated explicitly for the Log-logistic distribution but not for the Log-normal.

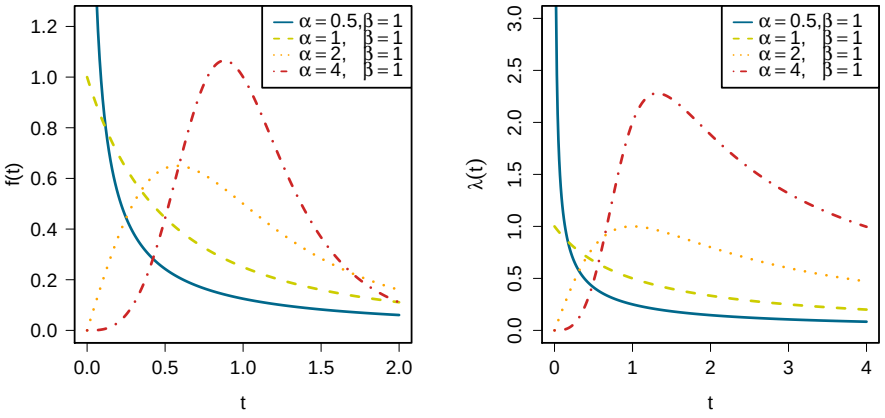
The corresponding cumulative hazard function is

$$\Lambda(t) = \log\left[1 + \left(\frac{t}{\beta}\right)^{\alpha}\right]$$

and the hazard function is

$$\lambda(t) = \frac{\alpha\beta^{-\alpha}t^{\alpha-1}}{1 + \left(\frac{t}{\beta}\right)^{\alpha}}.$$

The hazard function can have two different shapes depending on the value of the shape parameter α. When $\alpha \leq 1$ the hazard function is decreasing, while when $\alpha > 1$ it is hump-shaped (see Figure 2.6B).



(A) Density functions (B) Hazard functions

FIG. 2.6. Density and hazard functions of the Log-logistic distribution

2.7. Four-parameter Beta distribution

The *Four-parameter Beta distribution* [B(α, γ, a, b)] is a continuous distribution with two positive shape parameters (α and γ) and two parameters representing the minimum (a) and the maximum (b), these two last assumed to be known. This distribution is useful for fitting data which have an absolute maximum and minimum. Using the previous notation, the density function of the Beta distribution may be written as

$$f(t) = \frac{1}{B(\alpha, \gamma)} \frac{(t-a)^{\alpha-1}(b-t)^{\gamma-1}}{(b-a)^{\alpha+\gamma-1}}.$$

Being $B(\cdot,\cdot)$ the Beta function and $B_t(\cdot,\cdot)$ the incomplete Beta function

$$B(\alpha,\gamma) = \int_0^1 x^{\alpha-1}(1-x)^{\gamma-1}dx,$$

$$B_t(\alpha,\gamma) = \int_0^t x^{\alpha-1}(1-x)^{\gamma-1}dx,$$

the survival function may be written as

$$\begin{aligned} S(t) &= 1 - \frac{1}{B(\alpha,\gamma)} \int_0^t \frac{(x-a)^{\alpha-1}(b-t)^{\gamma-1}}{(b-a)^{\alpha+\gamma-1}}\, dx \\ &= 1 - \frac{B_{\frac{t-a}{b-a}}(\alpha,\gamma)}{B(\alpha,\gamma)} \\ &= \frac{B(\alpha,\gamma) - B_{\frac{t-a}{b-a}}(\alpha,\gamma)}{B(\alpha,\gamma)} \end{aligned}$$

The support of the four-parameter Beta distribution is the interval $[a,b]$ and both parameters α, γ must be positive. When $a = 0$ and $b = 1$, this distribution is known as the *standard beta distribution*. When the a and b limits are not specified [$\mathrm{B}(\alpha,\gamma)$], one will assume that we are referring to the standard beta.

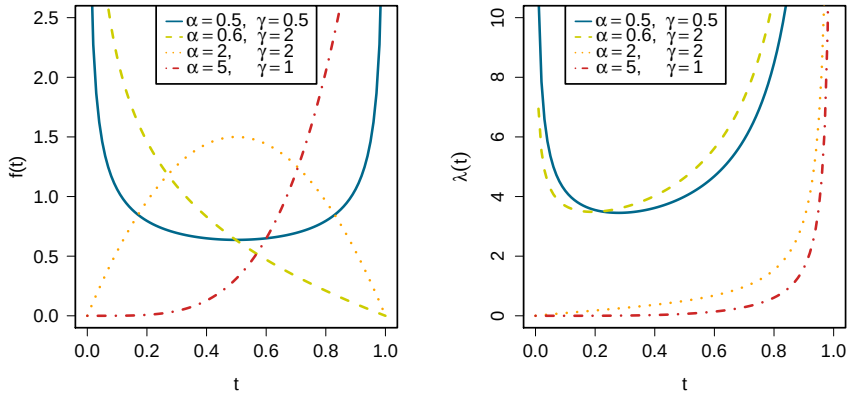
For the four-parameter Beta distribution, the cumulative hazard and the hazard function are, respectively,

$$\Lambda(t) = -\log\left(\frac{B(\alpha,\gamma) - B_{\frac{t-a}{b-a}}(\alpha,\gamma)}{B(\alpha,\gamma)}\right)$$

and

$$\lambda(t) = \frac{\left[(t-a)^{\alpha-1}(b-t)^{\gamma-1}\right]/(b-a)^{\alpha+\gamma-1}}{B(\alpha,\gamma) - B_t(\alpha,\gamma)}.$$

In Figures 2.7A and 2.7B the density and the hazard function for the standard Beta distribution ($a = 0$ and $b = 1$) are shown. Note that when $\alpha < 1$ the hazard is bathtub-shaped, but when $\alpha \geq 1$, it is increasing. Since a and b only affect the



(A) Density functions (B) Hazard functions

FIG. 2.7. Density and hazard functions of the standard Beta distribution

scale and the location of the distribution, the previous statement about the shape of the hazard function is true for any value of a and b.

2.8. Exponential power distribution

The *Exponential power distribution* [ExpPow(α, β)] was first introduced as a lifetime model by Smith and Bain (1975) [**SB75**]. In order to avoid confusions, we point out that, in statistical literature, it is possible to find the term "exponential power distribution" in a context that is not related with survival analysis but within asymmetrical distributions; see, for example, Delicado and Goria (2008) [**DG08**].

The Exponential power is a model that allows a bathtub shape for its hazard function and the expression of its survival function is rather simple. This is given by

$$S(t) = e^{1-e^{(\beta t)^\alpha}},$$

where $t > 0$, $\alpha > 0$ is the shape parameter and $\beta > 0$ is the scale parameter.

Its corresponding density function may be written as

$$f(t) = \alpha\beta^\alpha t^{\alpha-1} e^{(\beta t)^\alpha} e^{1-e^{(\beta t)^\alpha}},$$

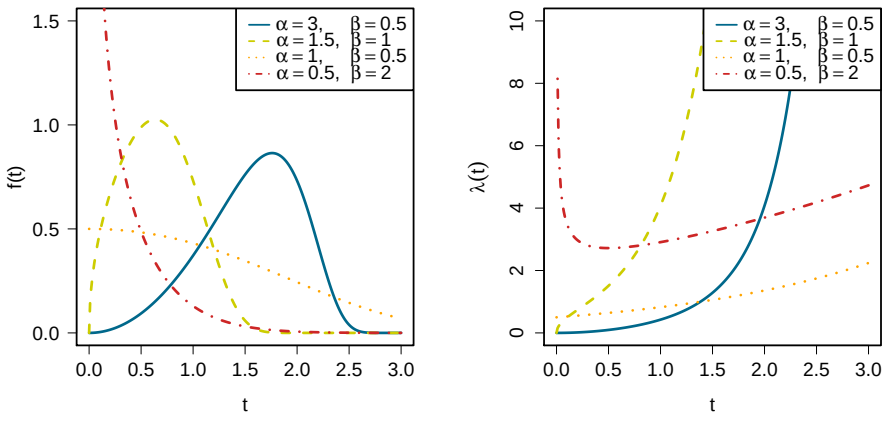
and the cumulative hazard function and the hazard function as

$$\Lambda(t) = e^{(\beta t)^\alpha} - 1$$

and

$$\lambda(t) = \alpha\beta^\alpha t^{\alpha-1} e^{(\beta t)^\alpha},$$

respectively.



(A) Density functions (B) Hazard functions

FIG. 2.8. Density and hazard functions of the Exponential Power distribution

Like the hazard function of the Beta distribution, the hazard of the Exponential power distribution can present two different shapes (increasing and bathtub-shaped) depending on parameter α. When $\alpha > 1$ the hazard function is increasing, while when $\alpha \geq 1$ it is bathtub-shaped (see Figure 2.8B).

2.9. Exponentiated Weibull distribution

The *Exponentiated Weibull distribution* [ExpWei(α, γ, β)] is a tri-parametric distribution with survival function given by

$$S(t) = 1 - \left[1 - e^{-(\beta t)^{\alpha}}\right]^{\gamma},$$

where $t > 0$, $\alpha > 0$ and $\gamma > 0$ are shape parameters and $\beta > 0$ is the scale parameter. The Exponentiated Weibull (α, γ, β) is a generalisation of the Weibull distribution, since for $\gamma = 1$ it represents the Weibull distribution with shape parameter α and scale parameter $1/\beta$.

Its density function is of the form

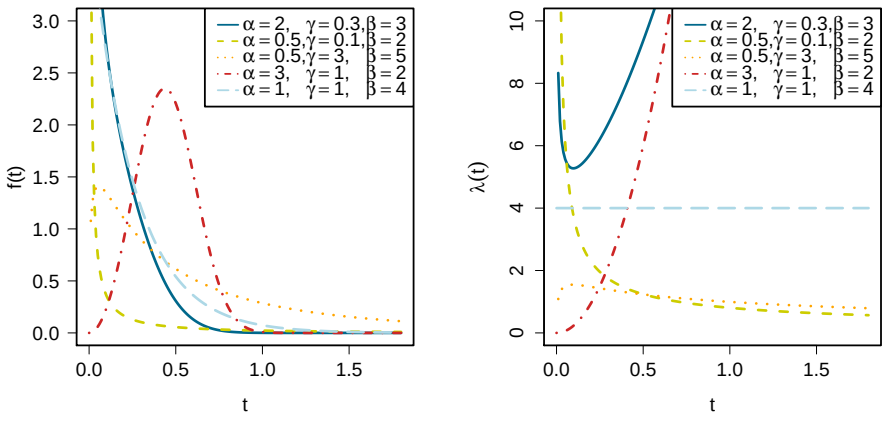
$$f(t) = \gamma\alpha\beta^{\alpha}t^{\alpha-1}e^{-(\beta t)^{\alpha}}\left[1 - e^{-(\beta t)^{\alpha}}\right]^{\gamma-1},$$

and the corresponding cumulative hazard and hazard function can be respectively expressed as

$$\Lambda(t) = -\log\left(1 - \left[1 - e^{-(\beta t)^{\alpha}}\right]^{\gamma}\right)$$

and

$$\lambda(t) = \frac{\alpha\beta\gamma(\beta t)^{\alpha-1}\left[1 - e^{-(\beta t)^{\alpha}}\right]^{\gamma-1}e^{-(\beta t)^{\alpha}}}{1 - \left[1 - e^{-(\beta t)^{\alpha}}\right]^{\gamma}}.$$



(A) Density functions (B) Hazard functions

FIG. 2.9. Density and hazard function of the Exponentiated Weibull distribution

Note that this hazard function can present the five possible shapes considered in this work – constant, increasing, decreasing, humped and bathtub-shaped (see Figure 2.9B). In this case, the shape of the hazard function depends not only on a single parameter but on two, which are α and γ:

- when $\alpha > 1$ and $\alpha\gamma \geq 1$ the hazard function is increasing,
- when $\alpha > 1$ and $\alpha\gamma < 1$ the hazard function is bathtub-shaped,
- when $\alpha < 1$ and $\alpha\gamma \geq 1$ the hazard function is hump-shaped,
- when $\alpha < 1$ and $\alpha\gamma < 1$ the hazard function is decreasing and
- when $\alpha = 1$ and $\alpha\gamma = 1$ the hazard function is constant.

2.10. Summary of the hazard rates of proposed distributions

As we have said in Section 1.2, the hazard rate is a very important function in survival and its graphic gives us a lot of information about the failure pattern. Hence we found it interesting to add a summarising table (see Table 2.1) pointing which types of behaviours can have the hazard function for each distribution. In the case that the hazard rate of a certain distribution can present more than one behaviour, it is indicated for which parameters values the hazard has one shape or another.

TABLE 2.1. Shape of the hazard function depending on the distribution and the parameters values.

		Shape				
		Increasing	Decreasing	Humped	Bathtub	Constant
Distributions	Wei(α, β)	$\alpha > 1$	$\alpha < 1$	–	–	$\alpha = 1$
	Gum(μ, β)	Always	–	–	–	–
	Norm(μ, β)	Always	–	–	–	–
	LNorm(μ, β)	–	–	Always	–	–
	Logis(μ, β)	Always	–	–	–	–
	LLogis(α, β)	–	$\alpha \leq 1$	$\alpha > 1$	–	–
	B(α, γ)	$\alpha \geq 1$	–	–	$\alpha < 1$	–
	ExpPow(α, β)	$\alpha \geq 1$	–	–	$\alpha < 1$	–
	ExpWei(α, γ, β)	$\alpha > 1$ $\alpha\gamma \geq 1$	$\alpha < 1$ $\alpha\gamma < 1$	$\alpha < 1$ $\alpha\gamma \geq 1$	$\alpha > 1$ $\alpha\gamma < 1$	$\alpha = 1$ $\alpha\gamma = 1$

Chapter 3
Assessing Goodness of Fit

Let us suppose that we have to analyse the data $(Y_1, \delta_1), (Y_2, \delta_2), \ldots, (Y_n, \delta_n)$; where $Y_i = \min(T_i, C_i)$ and δ_i is the censoring indicator. T_i is the random variable describing the event time, and $T_1, T_2, \ldots, T_n$ independent and identically distributed random variables with unknown cumulative distribution F. If we decide to perform parametric analysis on the data, an inappropriate theoretical distribution F_0 can lead to incorrect results. So it is important to validate if the data fits well or not to the chosen distribution. To asses this goodness of fit one needs to test the following hypothesis:

$$H_0 : F(\,\cdot\,) = F_0(\,\cdot\,;\boldsymbol{\theta})$$

where $F(\,\cdot\,)$ is the unknown distribution of the event times of our data and $F_0(\,\cdot\,,\boldsymbol{\theta})$ is the theoretical distribution that we want to adjust to the data and asses its goodness of fit. Note that this F_0 can be specified completely or up to some finite-dimensional parameter $\boldsymbol{\theta}$. When the parameters are not specified we might use the maximum likelihood estimate of $\boldsymbol{\theta}$, denoted by $\hat{\boldsymbol{\theta}}$. Let us denote $\hat{F}_0(t) = F_0(t;\hat{\boldsymbol{\theta}})$.

When we deal with uncensored data, F could be estimated by the empirical cumulative distribution function; but since our data present right-censored observations, F would be estimated by the Kaplan-Meier or the Nelson-Aalen estimators. This estimation will be denoted by $\hat{F}$.

In this Chapter some graphical and analytical methods for assessing the goodness of fit of a distribution to right-censored data will be introduced.

3.1. Probability Plots

The probability plots are useful tools to elucidate about if the chosen distribution is appropriate or not. These plots are useful to discard distribution that are clearly non valid.

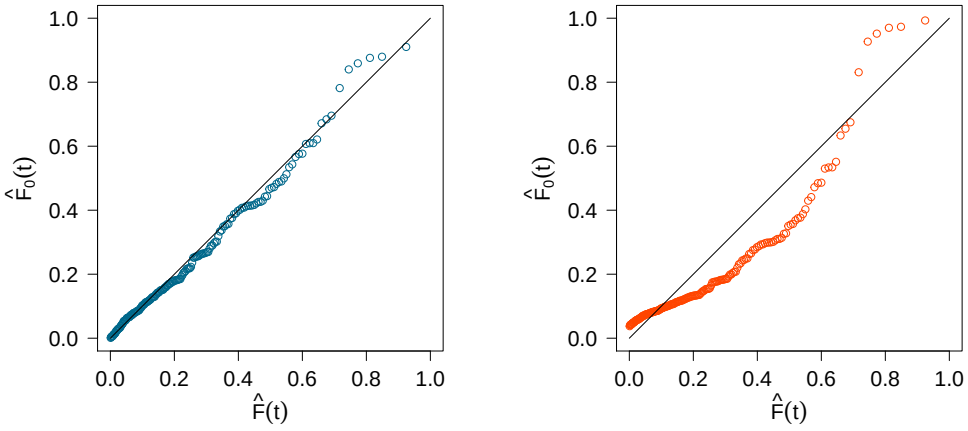
In this section we will introduce the most well-known probability plots – the P-P and the Q-Q plots – as well as two modifications of the P-P plot: the Stabilised Probability plot, which transforms the axes to approximately get the same variance in each plotted point and the Empirically Rescaled plot, which is very useful in the case that our data present a high percentage of random right-censored data.

To illustrate this plots we have simulated a sample set, of size 1000, from a Weibull(2,1) with an 82% of censored observations. For each plot we fitted a Weibull and a Gumbel distribution with the aim to show how the plot looks like when the distribution is appropriate and when it is not.

3.1.1. P-P plot

The Probability-Probability plot (or P-P plot) consists of plotting $\hat{F}_0(t)$ against $\hat{F}(t)$. That is, plotting the theoretical cumulative distribution function (with the parameters estimated by maximum likelihood if they are unknown) against the estimated cumulative distribution function derived from data. In the right-censored case, the estimation of the cumulative distribution function is computed using the Kaplan-Meier or the Nelson-Aalen estimators. The resulting graph must be a straight line from $(0, 0)$ to $(1, 1)$, if the data really follow the theoretical distribution. Otherwise, if the theoretical distribution does not fit the data, the resulting plot will be S shaped.

In Figure 3.1 we depict two plots to illustrate different resulting graphs in P-P plots. In Figure 3.1A we fitted a Weibull distribution, and note that the points are plotted all around the line, leading us to think that the Weibull distribution fits well the data (as it has been expected since the data was simulated from a Weibull). Otherwise, in Figure 3.2B, where we fitted a Gumbel distribution, the plotted points conform an S shaped figure. This S shaped form of the plotted points suggest us that the Gumbel distribution is not appropriate for the data.



(A) P-P plot when the distribution is appropriate

(B) P-P plot when the distribution is not appropriate

FIG. 3.1. Examples of P-P plots.

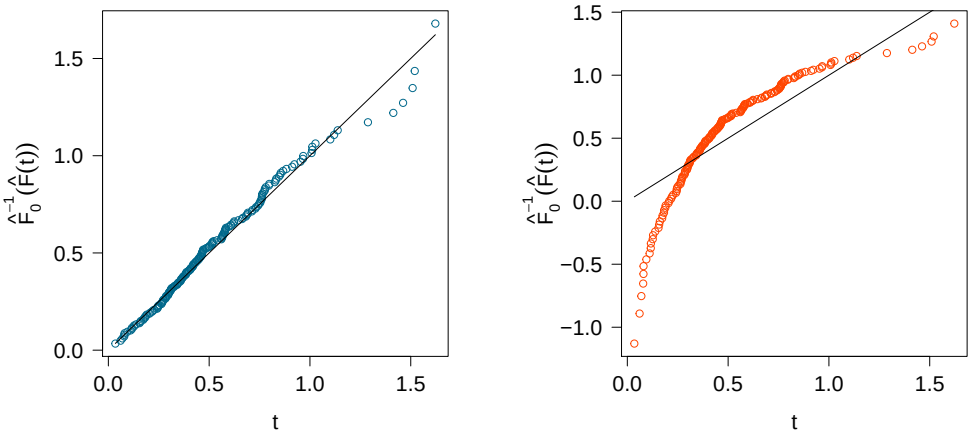
Note that the estimates of the distribution function F_0 only change at the uncensored observations, hence the points plotted in the P-P plot will only be the ones corresponding to the y_i when $\delta_i = 1$. When the data is uncensored or the censoring is of Type I or Type II, all the plotted points are evenly distributed over the

$(0,0)$ to $(\frac{r}{n},\frac{r}{n})$ line, where r is the number of observed events and n the sample size. Otherwise, when the data present a high proportion of random censored observations, since the censored points are not plotted, the evenly distribution of the plotted points over the line is no longer true. In this case, a tight group of points is plotted near the $(0,0)$, but as we approach to the other extreme of the line the points are more dispersed. This unevenly distribution of the plotted points over the line where the data set present a high proportion of random censored observations can be seen in Figures 3.1A and 3.1B, where 82% of the observations are randomly censored.

3.1.2. Q-Q plot

The Quartile-Quartile plot (or Q-Q plot) is similar to the P-P plot but this time the theoretical quartiles against the estimated quartiles are plotted. So the Q-Q plot consists of plotting $\hat{F}_0^{-1}(\hat{F}(t))$ against t. When the theoretical distribution F_0 fits well the data, the resulting plot will be a straight line; but if the distribution is not appropriate for the data, one will get a curved plot.

In Figure 3.2 we show two Q-Q plots, one adjusting a Weibull distribution (Figure 3.2A) and another adjusting a Gumbel distribution (Figure 3.2B). The plotted points in Figure 3.2A resemble a straight line pointing us that the Weibull distribution adjusts well to the data. Otherwise, the Q-Q plot showed in Figure 3.2B is curved and leads us to think that the Gumbel is not an appropriate distribution for the data.



(A) Q-Q plot when the distribution is appropriate

(B) Q-Q plotwhen the distribution is not appropriate

FIG. 3.2. Examples of Q-Q plots.

The Q-Q plot has the inconvenient that the plotted points may not be evenly spread, whether the data set presents censored observations or not. When the data set is large, most of the points would be concentrated in a narrow area, while a

handful of points would be spread over a wide region. For example, in Figures 3.2A and 3.2B the points are more concentrated around $t = 0.4$ and more dispersed in the extremes. The human eye is likely to give undue importance to the part of the plot with fewer points, leading to possibly biased conclusions.

3.1.3. Stabilised probability plot (SP plot)

The Stabilised Probability plot (or SP plot) was introduced by John R. Michael (1983) [**Mic83**] as a transformation of the P-P plot to stabilise the variance of the plotted points. That is, in this type of plot the variances of the plotted points are approximately equal. This fact is an attractive feature of the stabilised probability plot that enhances its interpretability.

The origin of the SP plots is that in the Q-Q plots and P-P plots some points have higher variance than others. For example, when the theoretical distribution is the normal distribution, in the Q-Q plots the points nearest to the centre of the graph have smaller variance than the points of the tails, while in the P-P plots the opposite happens.

When $F_0 = F$ and the parameters of F_0 are known, $\hat{F}_0(y_i)$ can be regarded as the realisation of a uniform order statistic. If the parameters of F_0 are unknown but efficiently estimated, then this is true asymptotically. Since the arcsin transformation can be used to stabilise the variance of a uniform order statistic, this transformation can stabilise the variance of $\hat{F}_0(y_i)$.

Suppose we let $S = \frac{2}{\pi} \arcsin\left(\sqrt{U}\,\right)$ where $U \sim \text{Uniform}[\,0, 1\,]$. Then the probability density function of S is given by

$$f(s) = \frac{\pi}{2} \sin(\pi s)$$

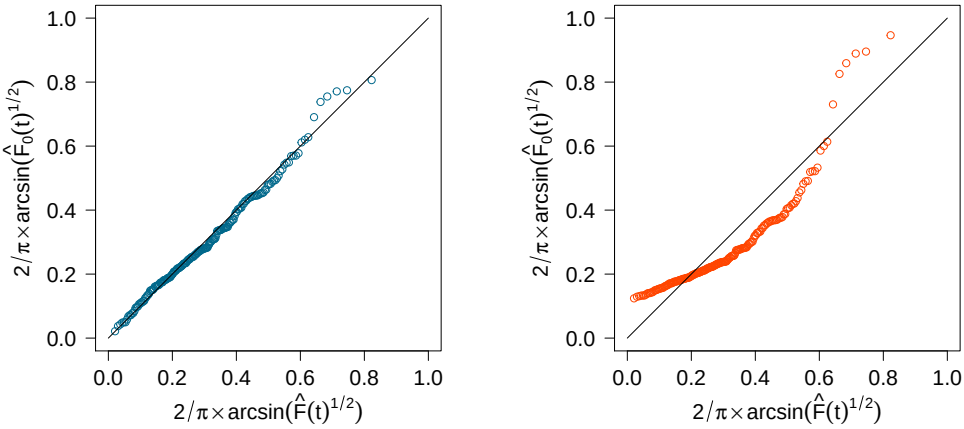
for $0 \leq s \leq 1$. This distribution is called the sine distribution and has the interesting property that its order statistics have the same asymptotic variance equals to $1/\pi^2$ independent of the order position.

The Stabilised Probability plot is defined as

$$\frac{\pi}{2} \arcsin\left(\sqrt{\hat{F}_0(y_i)}\right) \quad \text{vs.} \quad \frac{\pi}{2} \arcsin\left(\sqrt{\hat{F}(y_i)}\right).$$

If the distribution F_0 fits the data, then the resulting SP plot will be like the line from $(0, 0)$ to $(1, 1)$; whereas if the distribution F_0 is not appropriate for the data, the points will be plotted conforming an S shaped figure.

In Figure 3.3 we present some examples of the Stabilised Probability plots. In Figure 3.3A it is plotted the SP plot of adjusting a Weibull to the data and, as it is expected since the simulated data come from a Weibull, the points are plotted all around the continuous line. In Figure 3.3B we are adjusting a Gumbel and the plotted points look like an S. These two plot suggest us that the Gumbel distribution is not appropriate for our data while Weibull distribution do is.



(A) SP plot when the distribution is appropriate

(B) SP plot when the distribution is not appropriate

FIG. 3.3. Examples of Stabilised Probability plots.

As in P-P plots, when the data present a high proportion of random right-censored observations, the plotted points in the SP plot are not evenly distributed over the theoretical line. Since the censored observations do not appear in the plot and these observations usually correspond to large times, the space between the points near $(1, 1)$ will be larger that the space between the points near the origin. This phenomena can be seen in Figures 3.3A and 3.3B since the simulated data present an 82% of random censored observations.

3.1.4. Empirically rescaled plot (ER plot)

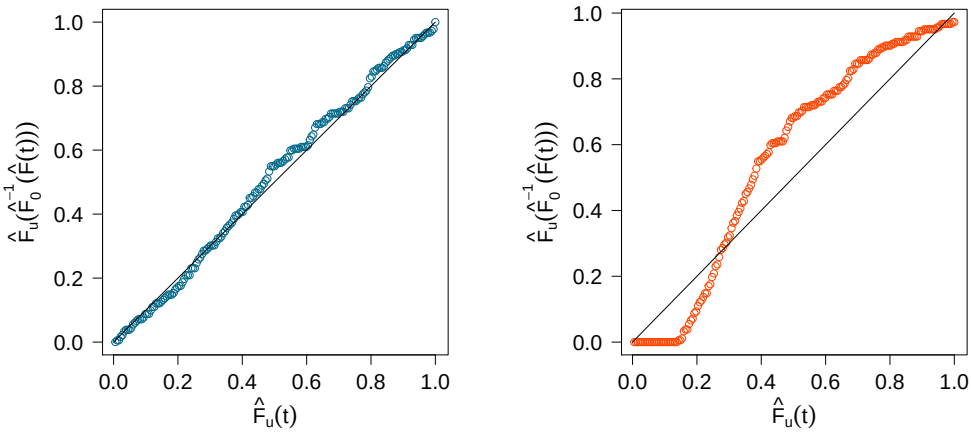
We mentioned before the inconvenient of the Q-Q plot presenting the plotted points not evenly spread. In the uncensored case or in the case that data present Type I or Type II censoring, this problem is partially solved using the P-P plot, since it has a uniform horizontal spacing between the points. However, this advantage is lost when the data is randomly right-censored, since the jumps of the empirical estimates of the probability function are not of the uniform size. Is for this reason that the Empirically Rescaled plot was proposed by . Waller and Turnbull (1992) [**WT92**].

The Empirically Rescaled plot consists of plotting

$$\hat{F}_u(\hat{F}_0^{-1}(\hat{F}(y_i))) \quad \text{vs.} \quad \hat{F}_u(y_i),$$

where $\hat{F}_u$ is the empirical cumulative distribution function of the points corresponding to the uncensored observations. If the theoretical distribution does not fit the data then the empirically rescaled plot will has a S shape, otherwise the plot will resemble the straight line from $(0, 0)$ to $(1, 1)$. Moreover its visual appearance is less sensitive to the effects of different censoring patterns than for the other plots considered.

In Figure 3.4A we adjusted a Weibull to the data, so it is an example of how the ER plot looks like when the adjusted distribution is appropriate for the data. In contrast, in Figure 3.4B we adjusted a Gumbel, a distribution that does not fit well to the data. Therefore the resulting Empirically Rescaled plot is not linear but S shaped as we mentioned above.



(A) ER plot when the distribution is appropriate

(B) ER plot when the distribution is not appropriate

FIG. 3.4. Examples of Empirically rescaled plots.

Note that in both Figures 3.4A and 3.4B, the points are evenly distributed despite the fact that the data present an 82% of random right-censored observations. We can say that the empirically rescaled plot fills a gap in existing goodness of fit graphical methods for large data sets with heavy random right-censoring.

3.2. Cumulative Hazard Plot

Similar to probability plots, cumulative hazard plots are used to visually asses the goodness of fit of a distribution to a data set. The cumulative hazard plot is based on transforming the cumulative hazard function Λ in such a way that it becomes linear in t or in $\log(t)$.

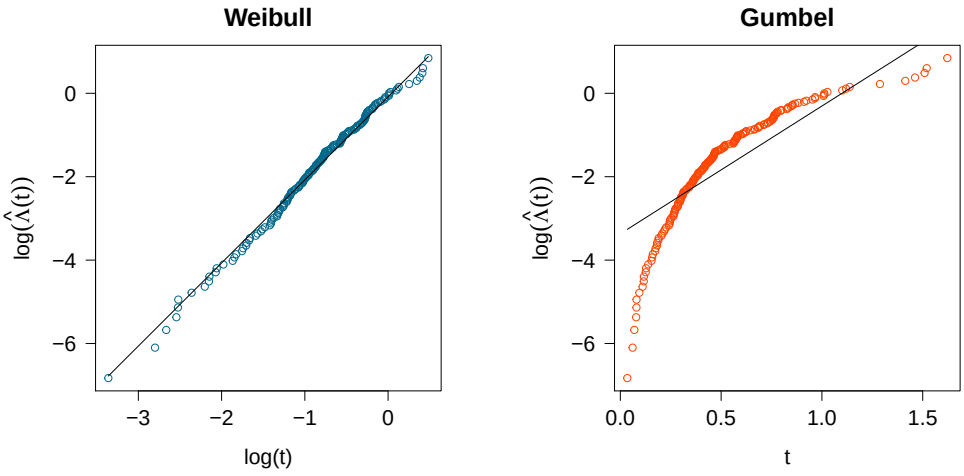
The transformation of the cumulative hazard function to build a cumulative hazard plot is specific for each distribution, but the idea behind is always the same. First of all we use the data to compute the Nelson-Aalen estimator, $\hat{\Lambda}$, of the cumulative hazard function and the maximum likelihood estimators of the parameters of the theoretical distribution we are adjusting. Then we look for a transformation $A(\,\cdot\,)$, related to the cumulative hazard function of the theoretical distribution, such that $A(\hat{\Lambda})$ will be linear in natural or logarithmic scale.

In Table 3.1 we present the specific expressions of the cumulative hazard plots for each of the nine distributions introduced in Chapter 2.

TABLE 3.1. Specific expressions for the Cumulative Hazard plot of each of the nine distributions.

Distribution	Cumulative Hazard, $\Lambda(t)$	Plot
Wei(α, β)	$(\beta t)^{\alpha}$	$\log \hat{\Lambda}(t)$ vs. $\log t$
Gum(μ, β)	$e^{\frac{t-\mu}{\beta}}$	$\log \hat{\Lambda}(t)$ vs. t
Norm(μ, β)	$-\log\left(1-\phi\left(\frac{t-\mu}{\beta}\right)\right)$	$\phi^{-1}\left(1-e^{-\hat{\Lambda}(t)}\right)$ vs. t
LNorm(μ, β)	$-\log\left(1-\phi\left(\frac{\log t-\mu}{\beta}\right)\right)$	$\phi^{-1}\left(1-e^{-\hat{\Lambda}(t)}\right)$ vs. $\log t$
Logis(μ, β)	$\log\left(1+e^{-\frac{t-\mu}{\beta}}\right)$	$\log\left(e^{\hat{\Lambda}(t)}-1\right)$ vs. t
LLogis(α, β)	$\log\left(1+\left(\frac{t}{\beta}\right)^{\alpha}\right)$	$\log\left(e^{\hat{\Lambda}(t)}-1\right)$ vs. $\log t$
B(α, γ)	$-\log\left(1-\frac{B_t(\alpha,\gamma)}{B(\alpha,\gamma)}\right)$	$F^{-1}_{\mathrm{B}(\alpha,\gamma)}\left(1-e^{-\hat{\Lambda}(t)}\right)$ vs. t
ExpPow(α, β)	$e^{(\beta t)^{\alpha}}-1$	$\log\left(\log\left(\hat{\Lambda}(t)+1\right)\right)$ vs. $\log t$
ExpWei(α, γ, β)	$-\log\left(1-\left(1-e^{-(\beta t)^{\alpha}}\right)^{\gamma}\right)$	$-\log\left(1-\left(1-e^{-\hat{\Lambda}(t)}\right)^{\frac{1}{\gamma}}\right)$ vs. $\log t$

To illustrate the resulting graphic of the Cumulative Hazard plot when the distribution fits the data and when not, we took the data simulated from a Weibull used in the previous section and we adjust to them a Weibull and a Gumbel distribution. In Figure 3.5A the Weibull distribution fits well the data and the points are plotted all around an straight line. In contrast, the plotted points in Figure 3.5B conform a curve, pointing the lack of fit of the Gumbel distribution to the data.



(A) Cumulative Hazard plot when the distribution fits the data.

(B) Cumulative Hazard plot when the distribution does not fit the data.

FIG. 3.5. Examples of Cumulative Hazard plots.

In the literature, sometimes the cumulative hazard plot is considered a probability plot consisting of plotting $\Lambda^{-1}(\Lambda(t))$ versus t. Using this definition, the cumulative hazard plot if the distribution is appropriate will always resembles the line from $(0,0)$ to $(1,1)$. Otherwise, our definition of cumulative hazard plot only want to get a line, it does not matter the scale, neither the slope nor the intercept. In practice, these two definitions work in the same way, but there is a subtle difference in their construction.

3.3. Grané Goodness of Fit test for Type I and Type II Right-censored Data

In Grané (2012) [**Gra12**] it is introduced a goodness of fit test that can be applied to data that present particular schemes of right, left and double censoring. Henceforth we will refer to this goodness of fit test as Grané test.

Let $T_1, \ldots, T_n$ be independent and identically distributed random variables with cumulative distribution function F and consider the order statistics $T_{(1)} < \ldots < T_{(n)}$. The Grané test is developed to test goodness of fit when the observations less than C_L and/or greater than C_R can not be observed (they are censored). Since in this work we are only considering right-censoring, we will focus on the case that only the time event values greater than C_R are the ones censored. Censoring may occur for fixed (Type I or time censoring) or for random values of C_R, Type I (or time censoring) and Type II (or failure censoring) respectively. So the Grané goodness of fit test is suitable for Type I and Type II censoring schemes but it can not be applied to random-censored samples.

The null hypothesis to be tested is

$$H_0 : F(t) = F_0(t) \quad \text{for every } t,$$

where F_0 is a completely specific cumulative distribution function. In practice we can only test it for the values $t_{(1)} < \ldots < t_{(r)}$ where r is the number of observed events. So what we are really testing is

$$H_0 : F(t_{(i)}) = F_0(t_{(i)}), \quad \text{for } i = 1, \ldots, r. \tag{3.1}$$

Remember that if T is a random variable with cumulative distribution function F, then $F(T)$ follows a Uniform distribution in $[0,1]$. Indeed,

$$P(F(T) < t) = P(T < F^{-1}(t)) = F(F^{-1}(t)) = t,$$

which corresponds to the cumulative distribution function of a Uniform in $[0,1]$.

With the above result, since T_i, for $i = 1, \ldots, n$, follows a distribution with cdf F, then $F(T_i)$ follows a Uniform in $[0,1]$. If the null hypothesis presented in (3.1) is true, $F_0(T_i)$ will also follow a Uniform in $[0,1]$. So, test if $F = F_0$ is equivalent to test that $F_0(t_{(i)})$ are iid random uniformly distributed in the $[0,1]$ interval, because if $F = F_0$, then $F_0(t_{(i)})$ will be a realisation of a Uniform $[0,1]$.

Note that the problem has been reduced to test the uniformity of $F_0(t_{(i)})$, for $i = 1, \ldots, r$. For convenience, from now we will denote $x_i = F_0(t_{(i)})$, $i = 1, \ldots, r$.

We can said that the Grané test is in fact a test for uniformity and is based on Hoeffding's maximum correlation coefficient introduced below.

Definition 3.1 (Hoeffding's maximum correlation coefficient) *Let F_1 and F_2 be two cumulative distribution functions with second order moments. The Hoeffding's maximum correlation coefficient between F_1 and F_2, henceforth denoted by $\rho^+(F_1, F_2)$, is defined as the maximum of the correlation coefficients of bivariate distributions having F_1 and F_2 as marginals:*

$$\rho^+(F_1, F_2) = \frac{1}{\sigma_1 \sigma_2} \left(\int_0^1 F_1^-(p) F_2^-(p) dp - \mu_1 \mu_2 \right), \tag{3.2}$$

where F_i^- is the left-continuous pseudo-inverse[1] *of F_i and, μ_i and σ_i^2 are, respectively, the expectation and the variance of F_i, $i = 1, 2$.*

The Hoeffding's maximum correlation coefficient $\rho^+(F_1, F_2)$ equals 1 if and only if $F_1 = F_2$ (almost everywhere) up to scale and location changes. Hence it is a measure of proximity between two distributions and yields a goodness of fit test statistic when replacing F_1 and F_2 by the empirical and hypothesised distributions.

In Fortiana and Grané (2003) [**FG03**] it has been studied, for complete samples, the test of uniformity based on

$$Q_n = \frac{s_n}{\sqrt{1/12}} \rho^+(F_n, F_U), \tag{3.3}$$

where F_n is the empirical cdf of n iid real-valued random variables, s_n is the sample standard deviation and F_U is the cdf of a uniform in $[0, 1]$ random variable. Later, in Grané (2012), the expressions of the modified Q_n statistic for type I and type II right-censored samples have been deduced and also their exact probability density function under the null hypothesis of uniformity. These deductions are presented below.

Before introducing the Q_n statistic for type I and type II right censoring, let us remark some points about the data. Let $t_{(1)} < \ldots < t_{(n)}$ be the ordered times. If the sample is right-censored of Type I, the times of the observed events $t_{(i)}$, $i = 1, \ldots, r$, are known to be less than a fixed value C_r, and the transformed x_i-values ($x_i = F(t_{(i)})$, $i = 1, \ldots, r$) also fulfil $x_{(1)} < \ldots < x_{(r)} < x^*$, where $x^* = F_0(C_R)$. This x^* will be also denoted as $x_{(r+1)}$. When the censoring is of Type II, there are again r values $x_{(i)}$, being $x_{(r)}$ the largest and r fixed.

Proposition 3.2 *Under the null hypothesis of uniformity:*

(i) The modified Q_n statistic for Type I right-censored data is

$$Q_{n_I} = \sum_{i=1}^{r+1} a_i x_{(i)},$$

where $a_i = \dfrac{6((2i-1)(r+1) - n^2)}{n^2(r+1)}$, *for* $1 \leq i \leq r$ *and* $a_{r+1} = \dfrac{6r(n^2 + r^2 - r)}{(n^2(r+1))}$.

[1] Suppose $F : \Omega_1 \to \Omega_2$ is a function with range $F(\Omega_1)$. A pseudo-inverse of F is a function $G : \Omega_2 \to \Omega_1$ that for all $x \in F(\Omega_1)$, $G(x)$ belongs to the preimage of x. The pseudo-inverse can be not unique.

(ii) The modified Q_n statistic for the Type II right-censored data is

$$Q_{n_{II}} = \sum_{i=1}^{r} a_i x_{(i)},$$

where $a_i = \dfrac{6((2i-1)r - n^2)}{n^2 r}$, *for* $1 \le i \le r-1$ *and* $a_r = \dfrac{6(r-1)(n^2 - r(r-1))}{n^2 r}$.

Proof. (i) For Type I right-censored data, let us suppose x^* ($x^* < 1$) is the fixed censoring value. This value can be added to the sample set, and the statistic can be calculated by using $x_{(r+1)} = x^*$. Note that it is possible to have $r = n$ observations less than x^*. In this case, when the value x^* is added then the new sample has size $n + 1$.
From formulae (3.2) and (3.3) we have that

$$\begin{aligned} Q_n &= \frac{s_n}{\sqrt{1/12}} \left[\frac{1}{s_n \sqrt{1/12}} \left(\int_0^1 F_n^-(p) F_U^-(p) dp - \frac{1}{2} \overline{x}_n \right) \right] \\ &= 12 \left(\int_0^1 F_n^-(p) F_U^-(p) dp - \frac{1}{2} \overline{x}_n \right) \end{aligned} \tag{3.4}$$

Noticing that the pseudo-inverse of the empirical cumulative distribution function is

$$F_n^-(p) = \begin{cases} x_{(i)}, & \frac{i-1}{n} < p \le \frac{i}{n}, \quad 1 \le i \le r, \\ x_{(r+1)}, & \frac{r}{n} < p \le 1, \end{cases}$$

for $0 \le p \le 1$, the first summand of (3.4) is

$$\begin{aligned} \int_0^1 F_n^-(p) F_U^-(p) dp &= \sum_{i=1}^{r} \int_{(i-1)/n}^{i/n} x_{(i)} p \, dp + \int_{r/n}^{1} x_{(r+1)} p \, dp \\ &= \frac{1}{2n^2} \sum_{i=1}^{r} (2i-1) x_{(i)} + \frac{1}{2n^2} (n^2 - r^2) x_{(r+1)} \end{aligned}$$

and subtracting the (available) sample mean and multiplying all by 12, the statistic Q_{n_I} is

$$Q_{n_I} = \sum_{i=1}^{r} \underbrace{\frac{6\left((2i-1)(r+1) - n^2\right)}{n^2(r+1)}}_{a_i} x_{(i)} + \underbrace{\frac{6r(n^2 - r^2 - r)}{n^2(r+1)}}_{a_{r+1}} x_{(r+1)}.$$

(ii) For Type II right-censored data, the pseudo-inverse of the empirical cumulative distribution function is

$$F_n^-(p) = \begin{cases} x_{(i)}, & \frac{i-1}{n} < p \le \frac{i}{n}, \quad 1 \le i \le r-1, \\ x_{(r)}, & \frac{r-1}{n} < p \le 1, \end{cases}$$

for $0 \le p \le 1$. Proceeding analogously, the first summand of (3.4) is

$$\int_0^1 F_n^-(p) F_U^-(p) dp = \sum_{i=1}^{r-1} \frac{(2i-1)}{2n^2} x_{(i)} + \left(\frac{1}{2} - \frac{(r-1)^2}{2n^2} \right) x_{(r)}$$

and subtracting the (available) sample mean and multiplying all by 12, the statistic $Q_{n_{II}}$ is

$$Q_{n_{II}} = \sum_{i=1}^{r-1} \underbrace{\frac{6((2i-1)r-n^2)}{n^2 r}}_{a_i} x_{(i)} + \underbrace{\frac{6(r-1)(n^2-r(r-1))}{n^2 r}}_{a_r} x_{(r)}$$

□

Under the null hypothesis, Q_{n_I} and $Q_{n_{II}}$ are linear combinations of selected order statistics from the $[0,1]$-uniform distribution. Therefore their exact probability density function can be obtained with the following algorithm proposed by Dwass (1961) [**Dwa61**], Matsunawa (1985) [**Mat85**] and Ramallingam (1989) [**Ram89**].

Mainly the algorithm for obtaining the pdf of Q_{n_I} and $Q_{n_{II}}$ is the same except from the computation of the b_i coefficients. For Type I right-censored data these coefficients are computed as

$$b_i = \sum_{l=i}^{r+1} a_l = \frac{6}{n^2}(2i - i^2 - 1) + \frac{6}{r+1}(i-1), \quad i = 1, 2, \ldots, r+1,$$

while for Type II right-censored data the b_i coefficients are

$$b_i = \sum_{l=i}^{r} a_l = \frac{6}{n^2}(2i - i^2 - 1) + \frac{6}{r}(i-1), \quad i = 1, 2, \ldots, r.$$

Once defined these coefficients, let k be the number of distinct non-zero b_i's and $(\nu_1, \ldots, \nu_k)$ be the corresponding multiplicities of $(b_1, \ldots, b_k)$. Defining on $\mathbb{C}$ the function:

$$G_l(s) = \left(s + \frac{1}{b_l}\right)^{\nu_l} \left[\prod_{j=i}^{k} \left(s + \frac{1}{b_j}\right)^{\nu_j}\right]^{-1}, \quad l = 1, 2, \ldots, k,$$

the exact pdf of Q_{n_I} and $Q_{n_{II}}$ statistics, under H_0, is given by

$$f(s) = \sum_{l=1}^{k} \sum_{m=1}^{\nu_l} \operatorname{sign}(b_l)\, C_{l,m}^{\sharp}\, \chi\left(\frac{s}{b_l}\right) \chi\left(1 - \frac{s}{b_l}\right) s^{m-1} \left(1 - \frac{s}{b_l}\right)^{n-m} \Big/ B(m, n-m+1)$$

where b_j are, respectively, the ones defined above for Type I and Type II right-censored, $\chi(x)$ is the indicator of the interval $[x > 0]$, $B(a,b)$ is the Beta function,

$$C_{l,m}^{\sharp} = \left(\prod_{j=1}^{k} b_j^{-\nu_j}\right) \frac{G_l^{(\nu_j - m)}\left(\frac{-1}{b_l}\right)}{(\nu_j - m)!},$$

and $G_l^{(j)}$ denotes the j-th derivative of G_l.

In Tables 3.2, 3.3, 3.4 and 3.5 there are presented the lower and upper-tail critical values for 5% and 2.5% significance levels (5% and 2.5% for each tail) of Q_{n_I} and $Q_{n_{II}}$ to test the null hypothesis for sample sizes up to 30 and different proportions p ($p = r/n$) of observed data in the sample.

TABLE 3.2. 5% and 95% critical values of Q_{n_I} for p proportions of observed events in the sample

p	$n = 10$		$n = 20$		$n = 30$	
0.3	0.2221	1.4577	0.3563	1.3467	0.4288	1.2751
0.4	0.3566	1.6455	0.5260	1.5568	0.6133	1.4925
0.5	0.4867	1.7344	0.6820	1.6820	0.7788	1.6315
0.6	0.5980	1.7228	0.8095	1.7167	0.9104	1.6853
0.7	0.6751	1.6095	0.8935	1.6554	0.9937	1.6465
0.8	0.6994	1.4005	0.9172	1.4956	1.0125	1.5103
0.9	0.6337	1.1368	0.8547	1.2518	0.9443	1.2876

TABLE 3.3. 2.5% and 97.5% critical values of Q_{n_I} for p proportions of observed events in the sample

p	$n = 10$		$n = 20$		$n = 30$	
0.3	0.1692	1.6314	0.3009	1.4778	0.3760	1.3828
0.4	0.2874	1.8059	0.4590	1.6805	0.5514	1.5952
0.5	0.4071	1.8725	0.6088	1.7917	0.7128	1.7239
0.6	0.5142	1.8326	0.7356	1.8076	0.8450	1.7631
0.7	0.5939	1.6875	0.8243	1.7240	0.9335	1.7067
0.8	0.6278	1.4500	0.8580	1.5414	0.9619	1.5516
0.9	0.5789	1.1735	0.8101	1.2817	0.9067	1.3886

TABLE 3.4. 5% and 95% critical values of $Q_{n_{II}}$ for p proportions of observed events in the sample

p	$n = 10$		$n = 20$		$n = 30$	
0.3	0.0999	1.1678	0.2714	1.2105	0.3659	1.1861
0.4	0.2221	1.4577	0.4418	1.4621	0.5531	1.4284
0.5	0.3566	1.6455	0.6066	1.6303	0.7267	1.5943
0.6	0.4867	1.7344	0.7503	1.7110	0.8712	1.6773
0.7	0.5980	1.7228	0.8579	1.6983	0.9722	1.6701
0.8	0.6751	1.6095	0.9141	1.5876	1.0145	1.5666
0.9	0.6994	1.4005	0.8993	1.3818	0.9789	1.3889

TABLE 3.5. 2.5% and 97.5% critical values of $Q_{n_{II}}$ for p proportions of observed events in the sample

p	$n=10$		$n=20$		$n=30$	
0.3	0.0684	1.3418	0.2240	1.3423	0.3172	1.2940
0.4	0.1692	1.6314	0.3800	1.5905	0.4938	1.5334
0.5	0.2874	1.8059	0.5359	1.7478	0.6615	1.6906
0.6	0.4071	1.8725	0.6760	1.8119	0.8051	1.7604
0.7	0.5142	1.8326	0.7857	1.7784	0.9098	1.7364
0.8	0.5939	1.6875	0.8493	1.6446	0.9602	1.6141
0.9	0.6278	1.4500	0.8469	1.4180	0.9367	1.4600

3.4. Kolmogorov-Smirnov Goodness of Fit test for Right-censored Data

The Kolmogorov-Smirnov goodness of fit test is the most used analytical method to test goodness of fit when one is dealing with non-censored data. In this case, the Kolmogorov-Smirnov statistic D_n for a given cumulative distribution function F_0 is defined by

$$D_n = \sup_t \mid F_0(t) - \hat{F}_n(t) \mid,$$

where $\hat{F}_n$ is the empirical distribution function of the data and n the data sample size. By the Glivenko-Cantelli theorem, if the sample comes from distribution F_0, then D_n converges to 0 almost surely when n goes to infinity. This result was carefully studied by Kolmogorov (1933) [**Kol33**] leading him to find the asymptotic distribution of D_n under the null hypothesis, known as Kolmogorov distribution. Some years later Smirnov (1939) [**Smi39**] studied the corresponding one-sided bounds, and in 1948 a table of this distribution was provided [**Smi48**].

The Kolmogorov distribution is the distribution of the random variable

$$K = \sup_{t\in[0,1]} \mid B(t) \mid$$

where $B(t)$ is the Brownian bridge and its cumulative distribution is given by

$$P(K \leq k) = 1 - 2\sum_{j=1}^{\infty}(-1)^{j-1}e^{-2j^2k^2}.$$

Under the null hypothesis that the sample comes from the hypothesised distribution F_0, Kolmogorov showed that

$$\sqrt{n}D_n \xrightarrow[n\to\infty]{\mathcal{D}} \sup_t \mid B(F_0(t)) \mid .$$

Moreover, if F_0 is continuous, then under the null hypothesis $\sqrt{n}D_n$ converges in distribution to the Kolmogorov distribution, which does not depend on F_0.

The Kolmogorov-Smirnov goodness of fit test is constructed by using the critical values of the Kolmogorov distribution, showed in Table 3.6. The null hypothesis is

rejected at level α if

$$\sqrt{n}D_n > K_\alpha$$

where K_α is found from

$$P(K \leq K_\alpha) = 1 - \alpha.$$

The asymptotic power of this test is 1, but in practice the statistic requires a relatively large number of data points to properly reject the null hypothesis.

TABLE 3.6. One-sided critical values for the Kolmogorov distribution.

	sample size n									
α	1	2	3	4	5	6	7	8	9	10
0.1	0.950	0.776	0.636	0.565	0.510	0.468	0.436	0.410	0.387	0.369
0.05	0.975	0.842	0.708	0.624	0.563	0.520	0.483	0.454	0.430	0.409
0.01	0.995	0.929	0.829	0.734	0.669	0.617	0.576	0.542	0.513	0.489
α	11	12	13	14	15	16	17	18	19	20
0.1	0.352	0.338	0.325	0.314	0.304	0.295	0.286	0.279	0.271	0.265
0.05	0.391	0.375	0.361	0.39	0.338	0.327	0.318	0.309	0.301	0.294
0.01	0.468	0.450	0.432	0.418	0.404	0.392	0.381	0.371	0.361	0.352
α	21	22	23	24	25	26	27	28	29	30
0.1	0.259	0.253	0.247	0.242	0.238	0.233	0.229	0.225	0.221	0.218
0.05	0.287	0.281	0.275	0.269	0.264	0.259	0.254	0.250	0.246	0.242
0.05	0.344	0.337	0.330	0.323	0.317	0.311	0.305	0.300	0.295	0.290
α	31	32	33	34	35	36	37	38	39	≥ 40
0.1	0.214	0.211	0.208	0.205	0.202	0.204	0.201	0.199	0.196	$1.224/\sqrt{n}$
0.05	0.238	0.234	0.231	0.227	0.224	0.226	0.223	0.220	0.217	$1.358/\sqrt{n}$
0.01	0.285	0.281	0.277	0.273	0.269	0.271	0.268	0.264	0.260	$1.628/\sqrt{n}$

In 1980, Fleming et al. [**FOOH80**] published a modification of the Kolmogorov-Smirnov test in an attempt to obtain increased power when applied to uncensored data. They also generalised this test for use with arbitrarily right-censored data. The steps they followed to get the modified Kolmogorov-Smirnov statistic are briefly explained below. But first of all let us introduce some additional notation:

Let $\{y_i : i = 1, \ldots, m\}$ be the set of m distinct and ordered observed times in the sample; that is, distinct times of death or censorship. Let $\{t_i : i = 1, \ldots, d\}$ be the subset of d distinct death times, and let $\{c_i : i = 1, \ldots, c\}$ be the subset of c distinct censorship times. Clearly $m \leq d+c \leq n$, being n the sample size. Let n_i represents the number of individuals that are at risk just before y_i. Furthermore, let d_i and l_i represent the number of individuals that fail and are censored, respectively, at time y_i. n_i, d_i and l_i can be extended to $n(s)$, $d(s)$ and $l(s)$, where $n(s)$ denotes the number of individuals that are at risk just before s, $d(s)$ represents the number of individuals that fail at time s and l_i is the number of individuals that are censored at time s.

Let us denote by S the survival function corresponding to times to death and by C the survival function associated with the censoring times. Let us also define $\Lambda = -\log S$ and $\Lambda_C = -\log C$.

Observe that in the non-censored case, the classical Kolmogorov-Smirnov statistic can be rewritten as:

$$\begin{aligned} D_n = \sup_t \mid F_0(t) - \hat{F}_n(t) \mid &= \sup_t \mid S_0(t) - \hat{S}_n(t) \mid \\ &= \sup_t \left| e^{-\hat{\Lambda}_n(t)} \left(e^{\hat{\Lambda}_n(t) - \Lambda_0(t)} - 1 \right) \right| \\ &= \sup_t \left| e^{-\hat{\Lambda}_n(t)} \int_0^t e^{\hat{\Lambda}_n(s) - \Lambda_0(s)} d\big[\hat{\Lambda}_n(s) - \Lambda_0(s)\big] \right| \\ &= \sup_t \left| \int_0^t \frac{\hat{S}_n(t) S_0(s)}{\hat{S}_n(s)} d\big[\hat{\Lambda}_n(s) - \Lambda_0(s)\big] \right| \end{aligned} \tag{3.5}$$

where S_0 and Λ_0 are the survival and the cumulative hazard function corresponding to the hypothesised distribution, and $\hat{S}_n = 1 - \hat{F}_n$ and $\hat{\Lambda}_n = -\log \hat{S}_n$, being $\hat{F}_n$ the empirical distribution function.

If $S_0(s) = S(s)$ for all $s \leq t$ almost everywhere, it follows that

$$\frac{S(t) S_0(s)}{S(s)} = \frac{1}{2}\big(S(t) - S_0(t)\big), \quad \text{almost everywhere.} \tag{3.6}$$

Glivenko and Cantelli (1933) [**Gli33**][**Can33**] proved that

$$\sup_t \mid F(t) - \hat{F}_n(t)| \longrightarrow 0 \ \text{ almost surely,}$$

where $\hat{F}_n$ is the empirical distribution function; therefore the equality in (3.6) is also true replacing $S(t)$ by $\hat{S}_n$.

From (3.5) and (3.6) and using the Glivenko-Cantelli theorem, the modification of the classical Kolmogorov-Smirnov statistic to test uncensored data proposed in Fleming et al. is derived:

$$\sqrt{n}\, \bar{D}_n(y_m) = \sup_{0 \leq t \leq y_m} \left| \frac{1}{2} \left(\hat{S}_n(t) + S_0(t) \right) \int_0^t \sqrt{n}\, \chi\big(N(s)\big) d\big[\hat{\Lambda}_n(s) - \Lambda_0(s)\big] \right|$$

being $\chi(x)$ the indicator of the interval $[x > 0]$. Fleming, O'Fallon, O'Brien and Harrington proved that

$$\sqrt{n}\, \bar{D}_n(y_m) \xrightarrow[n \to \infty]{\mathcal{D}} \sup_{0 \leq t \leq y_m} \mid B(F_0(t)) \mid$$

and that a test based in this statistic has higher power that the one based on the classical Kolmogorov-Smirnov statistic.

Note that for this modified Kolmogorov-Smirnov statistic, the changes in the cumulative hazard functions are weighted by the square root of the sample size. Indeed, when data present right-censored observations the amount of information available to estimate the change in the survival or cumulative hazard function at time s is only a fraction $C(s)$ of that available in uncensored data. Taking into account this fact, the modified Kolmogorov-Smirnov statistic can be generalised for use with

arbitrarily right-censored data. The goodness of fit procedure for censored data proposed by Fleming et al. is based on the following statistic:

$$\sqrt{n}\,\tilde{D}_n(y_m) = \sup_{0\le t\le y_m} \left| \frac{1}{2}\left(\hat{S}_n(t) + S_0(t)\right) \int_0^t \sqrt{n\,\hat{C}_n(s^-)}\,\chi\big(n(s)\big)\,d\big[\hat{\Lambda}_n(s) - \Lambda_0(s)\big] \right|$$

where $\hat{S}_n$ is the Nelson-Aalen estimator for the survival of the death times defined to break ties[2], $\hat{C}_n$ is the Nelson-Aalen estimator for the survival of the censoring times also defined to break ties, and $\hat{C}_n(s^-)$ is the left-hand limit of the function $\hat{C}_n$ at time s. Note that is statistic is valid because the Nelson-Aalen estimator for the survival is a consistent estimator of S and, under the null hypothesis, the equality shown in (3.6) is also true replacing S by the Nelson-Aalen survival estimator $\hat{S}_n$.

The statistic $\sqrt{n}\,\tilde{D}_n(y_m)$ also verifies that

$$\sqrt{n}\,\tilde{D}_n(y_m) \xrightarrow[n\to\infty]{\mathcal{D}} \sup_{0\le t\le y_m} |\,B(F_0(t))\,|\,.$$

The proof of this result is based on Lemma 3.3.

Lemma 3.3 *Let us denote*

$$\tilde{Y}_n = \frac{1}{2}\left(\hat{S}_{NA}(t) + S_0(t)\right) \int_0^t \sqrt{n\,\hat{C}_{NA}(s^-)}\,\chi\big(n(s)\big)d\big[\hat{\Lambda}_{NA}(s) - \Lambda_0(s)\big].$$

Then, when H_0 it true one has that

$$\{\tilde{Y}_n : 0 \le t \le y_m\} \xrightarrow[n\to\infty]{\mathcal{D}} \{B(F_0(t)) : 0 \le t \le y_m\},$$

where B is a time transformed Brownian bridge. Specifically, if $W = \{W(t) : t \ge 0\}$ represents Brownian motion, then

$$B(F(t)) = W(F(t)) - F(t)W(1).$$

Note that for $0 \le u \le t \le y_m$, $\operatorname{cov}\Big[B(F(u)), B(F(t))\Big] = F(u)[1 - F(t)]$.

Proof. It follows directly from results in Fleming and Harrington (1981) [**FH81**] that when the null hypothesis, $H_0 : F(t) = F_0(t)$ for all t, holds,

$$\left\{\int_0^t \sqrt{n\,\hat{C}_{NA}(s^-)}\,\chi\big(n(s)\big)d\big[\hat{\Lambda}_{NA}(s) - \Lambda_0(s)\big] : 0 \le t \le y_m\right\} \xrightarrow[n\to\infty]{\mathcal{D}} Z \equiv \{Z(t) : 0 \le t \le y_m\}\,,$$

where Z is a mean zero Gaussian process possessing continuous sample paths, independent increments, and variance function

$$\operatorname{var}\big(Z(t)\big) = \frac{F_0(t)}{1 - F_0(t)}.$$

Aalen (1976) [**Aal76**] proved that $\hat{\Lambda}_{NA}(t)$ is a uniformly strongly consistent estimator of $\Lambda(t)$ over the interval $[0, y_m]$, from which is straightforward to show that $\hat{S}_{NA}(t)$ possesses the same property with respect to $S(t)$. Corollary 1 of Theorem 5.1 in Billingsley (1968) [**Bil68**] states that, letting h be a measurable function and D_h the set of discontinuities of h, if the random variable X_n holds

$$X_n \xrightarrow[n\to\infty]{\mathcal{D}} X \qquad \text{and} \qquad P(X \in D_h) = 0,$$

[2] $\hat{S}_n = e^{-\hat{\Lambda}_{NA}(t)}$ where $\hat{\Lambda}_{NA}(t)$ is the Nelson-Aalen estimator for the cumulative hazard function defined in (1.4).

then

$$h(X_n) \xrightarrow[n\to\infty]{\mathcal{D}} h(X).$$

So, for this result, it follows that

$$\{Y_n(t) : 0 \le t \le t_m\} \xrightarrow[n\to\infty]{\mathcal{D}} \{S_0(t)Z(t) : 0 \le t \le t_m\}$$

since under the null hypothesis $\frac{1}{2}\big(\hat{S}_{NA}(t) + S_0(t)\big) \simeq S_0(t)$.

The proof of the lemma is completed by observing that, for $0 \le s \le t \le t_m$,

$$\operatorname{cov}\Big[S_0(s)Z(s), S_0(t)Z(t)\Big] = S_0(s)S_0(t)\operatorname{var}\big(Z(s)\big)$$

$$= S_0(s)S_0(t)\frac{F_0(s)}{1-F_0(s)} = F_0(s)[1-F_0(t)].$$

□

Schey (1977) [**Sch81**], in generalising the one-sided Kolmogorov goodness of fit test to the special case in which all observations are censored at the same point, provided the following lemma.

Lemma 3.4 *If B is a Brownian bridge over $[0,1]$ and $x \in (0,1)$, then*

$$P\left(\sup_{0<t<x} B(t) \ge y\right) = p(y,x)$$

where

$$p(y,x) = 1 - \phi\left(\frac{y}{\sqrt{x-x^2}}\right) + \phi\left(\frac{y(2x-1)}{\sqrt{x-x^2}}\right)e^{-2y^2}$$

with ϕ the standard normal cumulative function distribution. Moreover, for $p(x,y) \le 0.40$ we have

$$P\left(\sup_{0<t<x} |\,B(t)\,| \ge y\right) = 2p(y,x).$$

Combining results of Lemmas 3.3 and 3.4 one can obtain the asymptotic distribution of the Kolmogorov-Smirnov goodness of fit test statistic $\sqrt{n}\,\tilde{D}_n$. Hence, for large n, when the null hypothesis is true and $p(A, F(y_m)) \le 0.40$, being A the value of the statistic and $F(y_m)$ the image of the last recorded time, the $p-value$ can be approximated as follows

$$p-value = P\left(\sqrt{n}\,\tilde{D}_n(y_m) \ge A\right) \approx 2p\big(A, F(y_m)\big).$$

This approximation for the two-sided $p-value$, given by doubling $p\big(A, F(y_m)\big)$, was investigated by Schey (1977) [**Sch81**]. He found the approximation to be acceptable when $2p\big(y, F(y_m)\big) < 0.8$, and to be excellent when $2p\big(A, F(y_m)\big) < 0.2$. When $F(y_m) \ge 0.75$, e^{-2A^2} is an excellent approximation to $p(A, F(y_m))$, which will lead to slightly conservative estimates. If one chooses not to use these approximations, Koziol and Byar (1975) [**KB75**] have tabulated these two-sided $p-values$. In Table 3.7 there are presented the critical values of the $\sqrt{n}\,\tilde{D}_n$ for different levels of significance α (which its value corresponds to the $p-value$). Note that the numbers inside the table correspond to values of the statistic $\sqrt{n}\,\tilde{D}_n$.

TABLE 3.7. Critical values for the two-sided Kolmogorov-Smirnov statistic for right-censored data.

	$F(y_m)$									
α	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	1.00
0.990	0.1587	0.2232	0.2717	0.3115	0.3454	0.3747	0.3999	0.4209	0.4362	0.4410
0.975	0.1761	0.2473	0.3006	0.3441	0.3810	0.4125	0.4394	0.4612	0.4764	0.4806
0.950	0.1938	0.2718	0.3299	0.3771	0.4168	0.4504	0.4786	0.5011	0.5160	0.5196
0.900	0.2182	0.3054	0.3700	0.4219	0.4652	0.5014	0.5311	0.5540	0.5683	0.5712
0.850	0.2376	0.3321	0.4015	0.4571	0.5029	0.5409	0.5716	0.5946	0.6082	0.6106
0.800	0.2550	0.3559	0.4297	0.4883	0.5363	0.5756	0.6069	0.6300	0.6428	0.6448
0.750	0.2716	0.3785	0.4562	0.5176	0.5675	0.6079	0.6398	0.6626	0.6748	0.6764
0.700	0.2878	0.4005	0.4820	0.5460	0.5976	0.6391	0.6713	0.6938	0.7054	0.7067
0.650	0.3041	0.4225	0.5078	0.5743	0.6275	0.6699	0.7023	0.7245	0.7353	0.7365
0.600	0.3207	0.4449	0.5339	0.6029	0.6576	0.7008	0.7333	0.7551	0.7652	0.7662
0.550	0.3379	0.4681	0.5608	0.6322	0.6885	0.7323	0.7649	0.7863	0.7956	0.7964
0.500	0.3559	0.4923	0.5889	0.5627	0.7204	0.7649	0.7975	0.8183	0.8270	0.8276
0.450	0.3750	0.5180	0.6185	0.6949	0.7541	0.7992	0.8316	0.8518	0.8597	0.8602
0.400	0.3956	0.5455	0.6503	0.7293	0.7899	0.8356	0.8678	0.8872	0.8944	0.8948
0.350	0.4181	0.5755	0.6849	0.7666	0.8287	0.8748	0.9068	0.9254	0.9318	0.9321
0.300	0.4431	0.6088	0.7231	0.8078	0.8715	0.9180	0.9496	0.9673	0.9729	0.9731
0.250	0.4714	0.6465	0.7663	0.8544	0.9196	0.9666	0.9976	1.0142	1.0190	1.0192
0.200	0.5045	0.6905	0.8168	0.9085	0.9756	1.0229	1.0533	1.0687	1.0727	1.0727
0.150	0.5449	0.7443	0.8784	0.9746	1.0438	1.0914	1.1208	1.1348	1.1379	1.1379
0.100	0.5935	0.8155	0.9597	1.0616	1.1334	1.1813	1.2094	1.2216	1.2238	1.2238
0.050	0.6825	0.9268	1.0868	1.1975	1.2731	1.3211	1.3471	1.3568	1.3581	1.3581
0.025	0.7589	1.0282	1.2024	1.3209	1.3997	1.4476	1.4717	1.4794	1.4802	1.4802
0.010	0.8512	1.1505	1.3419	1.4696	1.5520	1.5996	1.6214	1.6272	1.6276	1.6276
0.005	0.9157	1.2361	1.4394	1.5735	1.6583	1.7056	1.7258	1.7306	1.7308	1.7308
0.001	1.0523	1.4171	1.6456	1.7931	1.8828	1.9292	1.9464	1.9494	1.9495	1.9495

Next we will define the procedure proposed by Fleming et al. to calculate the generalised Kolmogorov-Smirnov goodness of fit test based upon the statistic $\sqrt{n}\,\tilde{D}_n(y_m)$:

(i) Set $\hat{\Lambda}_n(y_0) = \hat{\Lambda}_{nC}(y_0) = 0$ and recursively calculate, for $i = 1, \ldots, m$,

$$\hat{\Lambda}_n(y_i) = \hat{\Lambda}_n(y_{i-1}) + \sum_{k=0}^{d_i-1} \frac{1}{n_i - k}$$

and

$$\hat{\Lambda}_{nC}(y_i) = \hat{\Lambda}_{nC}(y_{i-1}) + \sum_{k=0}^{l_i-1} \frac{1}{n_i - d_i - k}.$$

This recursively procedure basically computes the Nelson-Aalen estimator for the cumulative hazard function defined in (1.4) adopting the convention of breaking the ties between deaths and censorships assuming that the deaths occurred infinitesimally earlier. Set $\hat{S}_n(t) = e^{-\hat{\Lambda}_n(t)}$ and $\hat{C}_n(t) = e^{-\hat{\Lambda}_{nC}(t)}$.

(ii) Setting $A(y_0) = B(y_0) = 0$, recursively calculate, for $i = 1, \ldots, m$,

$$A(y_i) = A(y_{i-1}) + \sqrt{\hat{C}_n(y_{i-1})} \log\left(S_0(y_{i-1})/S_0(y_i)\right)$$

and

$$B(y_i) = B(y_{i-1}) + \sqrt{\hat{C}_n(y_{i-1})} \sum_{k=0}^{d_i-1} \frac{1}{n_i - k}.$$

(iii) For $i = m$ and for all i such that $y_i \in \{t_1, \ldots, t_d\}$, calculate

$$Y_n(y_i^-) = \frac{1}{2}\sqrt{n}\left(\hat{S}_n(y_{i-1}) + S_0(y_i)\right)\left(A(y_i) - B(y_{i-1})\right)$$

and

$$Y_n(y_i) = \frac{1}{2}\sqrt{n}\left(\hat{S}_n(y_i) + S_0(y_i)\right)\left(A(y_i) - B(y_i)\right).$$

(iv) Set

$$A = \max\left\{|Y_n(t_i^-)|, |Y_n(t_i)|, |Y_n(y_m)| : i = 1, \ldots, d\right\},$$

which corresponds to the value of the statistic and set

$$R = 1 - \frac{1}{2}\left(\hat{S}_n(y_m) + S_0(y_m)\right),$$

which can be seen as an estimation of $F(y_m)$ under the null hypothesis.

(v) Finally calculate the $p - value$ for the test, when $p(A, R) < 0.40$, with the following formula

$$p - value = 2p(A, R) = 2\left(1 - \phi\left(\frac{A}{\sqrt{R - R^2}}\right) + \phi\left(\frac{A(2R-1)}{\sqrt{R - R^2}}\right)e^{-2A^2}\right).$$

Chapter 4
Tools for assessing Goodness of Fit for Right-censored data

The four methods to assess Goodness of Fit introduced in Chapter 3 had been implemented in `R` aiming to create a local library with functions to test goodness of fit for right-censored data.

We built four functions, one per method, called `prob.plots`, `CumHazPlot`, `KScens` and `Grane.test`. The goal of this Chapter is to introduce them, to show how one can call them, which input arguments are needed for each function and which output we will get. We also used this Chapter to explain how these four functions internally work.

The idea behind each function is that given survival data and being specified a theoretical distribution F_0, use the data in such a way that at a the end one can suggest whether or not the distribution F_0 is appropriate to the data. Our functions consider the nine different distributions introduced in Chapter 2 as the theoretical distribution F_0. Among these nine distributions we can find symmetric, right and left skewed distributions, and each of the five common shapes for the hazard rate is present in at least one of these distribution. These nine distributions cover a large range of parametrical families

The `R` code of each of these functions can be found in Appendices A to D.

4.1. Probability plots – `prob.plots` function

In Section 3.1 we introduced four types of probability plots: the P-P plot, the Q-Q plot, the Stabilised probability plot and the Empirically Rescaled plot. Since SP plots and ER plots are kind of improvements of the P-P and Q-Q plots, specially ER plots when dealing whit right-censored data, we thought that the image of the four plots together can help to point if the data fit well to the tested theoretical distribution or not. Based on this idea we built the `prob.plots` function.

4.1.1. Usage and input arguments

The usage of the `prob.plots` function is the following

```
prob.plots (time, cens, distributions, beta.limits = c(0,1),
            plots = c("PP","QQ","SP","ER"),
            colour = c("green4","deepskyblue4",
                       "yellow3","mediumvioletred"),
            parameters = list(shape = NULL, shape2 = NULL,
                              location = NULL, scale = NULL))
```

The `prob.plots` function has seven input arguments, which correspond to

`time`	The vector of times until the studied event
`cens`	The vector indicating the censored observations
`distribution`	An string specifying the name of the distribution to be studied. The possible distributions are the exponential (`"exp"`), the weibull (`"weibull"`), the gumbel (`"gumbel"`), the normal (`"norm"`), the log-normal (`"lnorm"`), the logistic (`"logis"`), the log-logistic (`"loglogis"`), the beta (`"beta"`), the exponential power (`"exppower"`) and the exponentiated weibull (`"expweibull"`).
`beta.limits`	A two components vector corresponding to the lower and upper bounds of the Beta distribution. This argument is only required if the beta distribution is considered. By default, it is set to `c(0,1)`.
`plots`	A vector stating the plots to be displayed. `"PP"` corresponds to the P-P plot, `"QQ"` to the Q-Q plot, `"SP"` corresponds to the Stabilised Probability plot and `"ER"` to the Empirically Rescaled plot. By default the four plots are displayed.
`colour`	A vector indicating the colours that each of the displayed plots must be painted. This vector is used cyclically; that is, if its length is smaller than the number of plots to be displayed, after being used the last colour we will use again the first one.
`parameters`	A list specifying the parameters of the theoretical distribution. By default they are set to `NULL` and estimated with the maximum likelihood estimate. This argument is only considered if all parameters of the studied distribution are specified.

4.1.2. Output

By default, the output of the `prob.plots` function is an image with the four probability plots studied in Chapter 3. To illustrate it let us take a simulated survival

data set of size 300 coming from a Weibull$(2, 1)$ with a 50% of censored observations. Let `time` be the vector of measured times and `cens` the vector of the censoring indicators. So, with the line

```
prob.plots(time, cens, "weibull")
```

we are testing the goodness of fit of the Weibull distribution, and the graphical output we get from the function is showed in Figure 4.1.

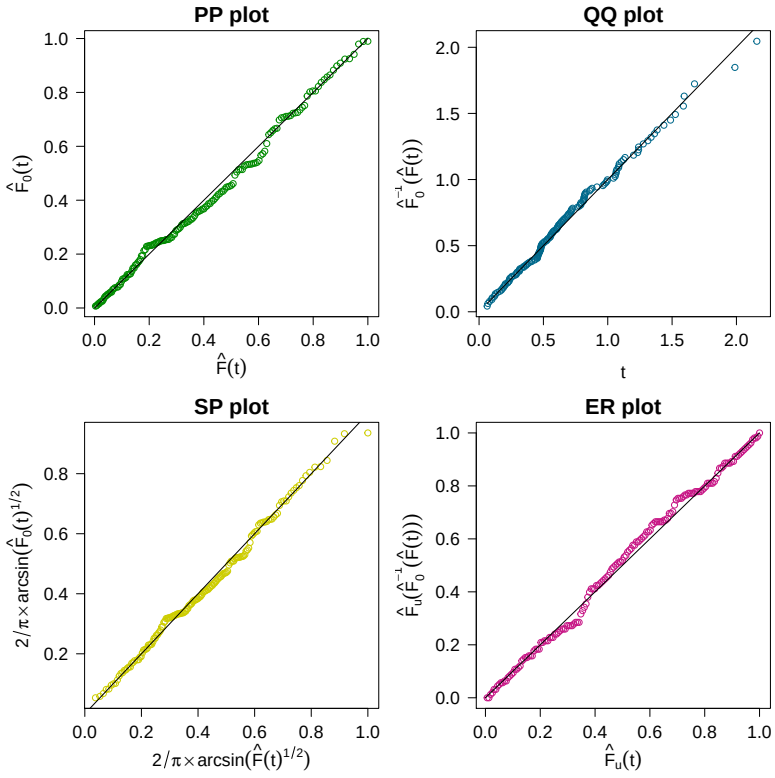


FIG. 4.1. Example of the `prob.plots` output.

Apart from the graphical output, the function also returns a list pointing which distribution we are testing and the value of its parameters, corresponding to the ones introduced by the user or to their maximum likelihood estimates if they are unknown. Following with the example, what we get in the `R` console is

```
$distr
[1] "Weibull"

$shape
[1] 1.827145

$scale
[1] 0.9466134
```

The style of the previous output is the one obtained by default, but the user can choose which probability plots want to be displayed and change the colour of each of them. For example, let us suppose that we want to assess the goodness of fit of the Exponential power distribution to the data but we are only interested in the P-P plot and the ER plot and we want them to be displayed in tones of blue. We can get this desired figure using the following line

```
prob.plots(time, cens, "exppower", plots=c("PP","ER"),
           colour = c("deepskyblue4", "cadetblue2"))
```

and the output will be the one displayed in Figure 4.2

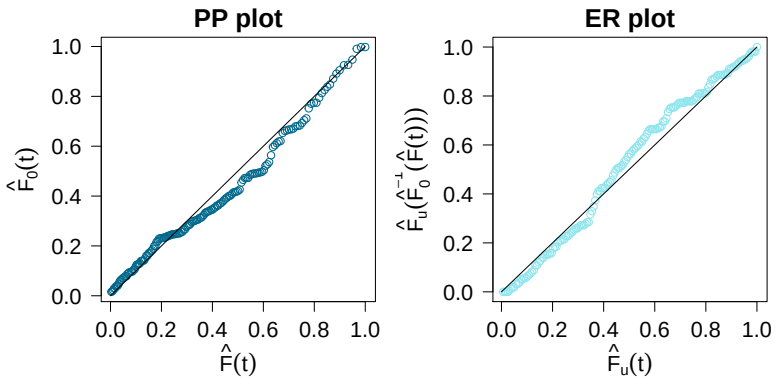


FIG. 4.2. Example of a customised `prob.plots` output, changing the number of displayed plots and their colour.

4.1.3. How does it work?

The `prob.plots` function takes the introduced survival data and use them to give some probability plots with the aim to assess the goodness of fit of the considered theoretical distribution. To do that, the function needs to estimate the parameters of the considered distribution (except when these have been specified by the user) and, for each plot, to transform the data in a way to get a resulting linear graph if the considered distribution is appropriate for the data. Next, some details are provided.

Estimation of distribution parameters

The hypothesis that we want to test with the `prob.plots` function is

$$H_0 : F = F_0(\cdot\,; \boldsymbol{\theta}),$$

where F is the real cdf of the data and F_0 the cdf of the distribution specified in the argument `distribution`. If the parameters of F_0 are indicated in argument `parameters`, we can skip this step since the distribution we are testing is completely defined. Otherwise, its parameters must be estimated, and we chose to do it using the maximum likelihood estimator.

There exists an R package called `fitdistrplus` [**DMD14**] that contains a function that fit a univariate distribution to censored data by maximum likelihood. This function is called `fitdistcens` and we used it to estimate the parameters of the considered distribution.

Constructing the Probability plots

Once the distribution is completely specified, with specific values for its parameters, this is used to transform the data with the purpose to get a linear plot if the distribution fits well to the data. Each probability plot transforms the data in a different way, but all these transformation lead to a linear graph if the distribution is appropriate and to a curved one if the distribution fails in fitting well to the data.

In Table 4.1 there are showed the transformations that each plot applies to the data to get a linear plot.

TABLE 4.1. Probability plots for testing goodness of fit.

Plot	Abscissa	Ordinate
P-P plot	$\hat{F}(y_i)$	$\hat{F}_0(y_i)$
Q-Q plot	y_i	$\hat{F}_0^{-1}(\hat{F}(y_i))$
S-P plot	$\frac{\pi}{2}\arcsin\left(\hat{F}(y_i)^{\frac{1}{2}}\right)$	$\frac{\pi}{2}\arcsin\left(\hat{F}_0(y_i)^{\frac{1}{2}}\right)$
E-R plot	$\hat{F}_u(y_i)$	$\hat{F}_u(\hat{F}_0^{-1}(\hat{F}(y_i)))$

* The notation is the one used in Chapter 3.

4.2. Cumulative Hazard plots – `CumHazPlot` function

It is believed that graphical methods are better than analytical ones in order to validate the goodness of fit of a distribution to certain data. This is because the goodness of fit tests have very low power for small and moderate sample size or they tend to reject any model for large sample sizes. So, being aware of this fact, we decided to build a function that let us compare graphically how data fits to different distributions. We named this function `CumHazPlot` .

The idea behind the `CumHazPlot` is to visually compare at once several distributions for a given data set. For a given survival data up to nine cumulative hazard plots, corresponding to different distributions including symmetric, left and right skewed and with increasing, decreasing, constant, humped and bathtub shaped hazard function, are depicted. Based on these plots, the user can decide which distribution choose for the analysis.

4.2.1. Usage and input arguments

The usage of the `CumHazPlot` function is the following:

```
CumHazPlot (time, cens,
            distributions = c("gumbel","norm","logis",
                              "weibull","lnorm","loglogis"),
            beta.limits = c(0,1),
            colour = c("orangered","darkolivegreen3","cadetblue2",
                       "red3","green4","deepskyblue4",
                       "hotpink","yellow3","mediumvioletred"))
```

Note that the function has five input arguments, which correspond to

`time`	The vector of times until the studied event
`cens`	The vector indicating the censored observations
`distributions`	A vector with the names of the distributions to be studied. The possible distributions are the weibull (`"weibull"`), the gumbel (`"gumbel"`), the normal (`"norm"`), the log-normal (`"lnorm"`), the logistic (`"logis"`), the log-logistic (`"loglogis"`), the beta (`"beta"`), the exponential power (`"exppower"`) and the exponentiated weibull (`"expweibull"`). The option `"all"` takes into consideration the nine distributions. By default this argument is set to `c("gumbel","norm","logis","weibull","lnorm", "loglogis")`, which are the most used distributions in survival.
`beta.limits`	A two components vector corresponding to the lower and upper bounds of the Beta distribution. This argument is only required if the beta distribution is considered. By default, it is set to `c(0,1)`.
`colour`	A vector indicating the colours that each of the displayed plots must be painted. This vector is used cyclically; that is, if its length is smaller than the number of plots to be displayed, after being used the last colour we will use the first one again and so on.

4.2.2. Output

By default, the output of the `CumHazPlot` consists in the six cumulative hazard plots corresponding to the Gumbel, the Normal, the Logistic, the Weibull, the Log-normal and the Log-logistic distributions. To illustrate this output let us take a simulated survival data set of size 250 coming from the Standard Lognormal distribution with a censoring percentage of 30%. Let `time` be the vector of times until the event and `cens` the vector containing the censoring indicator for each failure time. Hence, calling the function

```
CumHazPlot(time, cens)
```

we will get and output like the one showed in Figure 4.3.

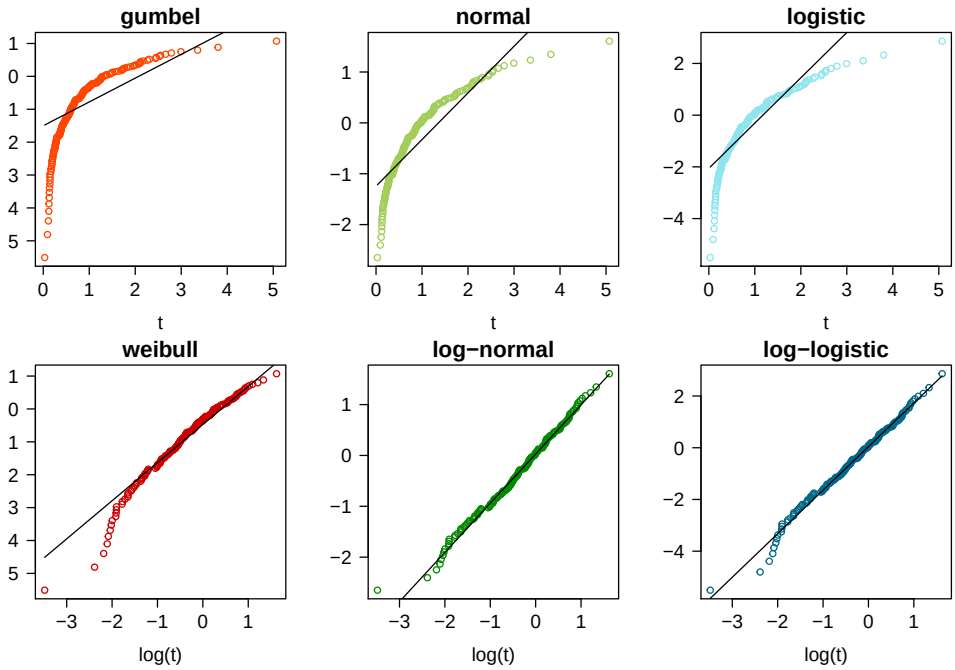


FIG. 4.3. Example of the CumHazPlot output when the data comes from a log-normal.

Each figure shows the cumulative hazard plot, as well as the theoretical line. This output is a visual aid to decide whether or not the data fit enough well to the corresponding theoretical distributions. The closer the coloured curve is to the straight line, the better the data fit to the distribution.

From the plot the user can decide which distribution prefers to use for the subsequent analysis, but once decided the distribution one want to know which parameter values to use. It is for this reason that we decided to provide a list with the maximum likelihood estimates of the parameters for each considered distribution.

In our example we have considered six distributions (Gumbel, Normal, Logistic, Weibull, Lognormal and Loglogistic) and we got the following console output:

```
$gumbel                          $weibull
 location      scale                shape      scale
 2.075058   1.382225               1.158071   1.506661

$normal                          $lognormal
 location      scale                location      scale
 1.357758   1.090613             -0.0288776   1.0334896
```

```
$logistic                          $loglogistic
 location      scale                    shape       scale
1.1750172  0.5709292                1.6838615   0.9740563
```

The default output has the six cumulative hazard plots depicted in Figure 4.3, but the number of displayed plots can be modified. In fact the `CumHazPlot` output will show as many plots as the indicated in the `distributions` argument. The output can also be customised by changing the colours of the plots.

If we try to fit a distribution to data that exceed the distribution domain the function will return an image like the one showed in Figure 4.4

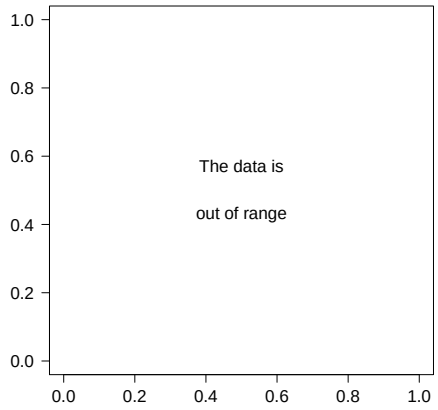


FIG. 4.4. Output when the data exceed the domain of the fitting ditribution.

4.2.3. How does it work?

The `CumHazPlot` function provides the cumulative hazard plots for the survival data (the vector of failure times and the vector of censoring indicators) introduced by the user. The function estimates the parameters for each distribution and computes the estimated cumulative hazard. A transformation of the units, of the cumulative hazard or of both of them is performed and a plot is drawn. If the considered distribution is appropriate for the data, the resulting plot will be approximately linear.

Now we give more details of how these steps are carried out.

Estimation of the cumulative hazard function

A survival object is created via the `R` function `Surv` of the `survival` package from the failure times and the censoring indicators vector. This survival object is used to compute the Nelson-Aalen estimator for the cumulative hazard function, $\hat{\Lambda}_{NA}$. By means of the `survfit` function, the derived survival estimator is taken as an argument and the Nelson-Aalen estimator is computed.

Estimation of distribution parameters

The `CumHazPlot` tests, for each of the indicated distributions in the `distributions` argument, the hypothesis:

$$H_0 : F = F_0,$$

where F is the real distribution the data come from and F_0 the considered distribution in each case. Note that the parameters $\boldsymbol{\theta}$ of the F_0 distribution are not specified. So we estimate $\boldsymbol{\theta}$ by the maximum likelihood estimator, $\hat{\boldsymbol{\theta}}_{MLE}$, and in fact we test the hypothesis

$$H_0 : F(t) = F_0(t; \hat{\boldsymbol{\theta}}_{MLE}).$$

As in `prob.plots` , we use the `fitdistcens` function to compute the maximum likelihood estimates for the parameters of the considered distributions.

Constructing the Cumulative Hazard plots

From the estimated cumulative hazard function, for a given distribution, and from the corresponding maximum likelihood estimated parameters, we transform the cumulative hazard in such a way that the resulting plot will be linear (in the natural or logarithmic scale) if the data fit well to the distribution $F_0(\,\cdot\,; \hat{\boldsymbol{\theta}}_{MLE})$. In Table 4.2 we present the explicit transformation of the estimated cumulative hazard $\hat{\Lambda}$ for each of the nine considered distributions and the scale that will make the plot of the transformed cumulative hazard linear if the data fits the distribution.

TABLE 4.2. Transformations of the estimated cumulative hazard and the corresponding scale to get a linear plot under the null hypothesis. ϕ is the cdf of a Normal$(0, 1)$ and $F_{\mathrm{B}(\alpha,\beta)}$ the cdf of a standard Beta.

Distribution	Transformation	Scale
$\mathrm{Wei}(\alpha, \beta)$	$\log \hat{\Lambda}(t)$	logarithmic
$\mathrm{Gum}(\mu, \beta)$	$\log \hat{\Lambda}(t)$	natural
$\mathrm{Norm}(\mu, \beta)$	$\phi^{-1}\left(1 - e^{-\hat{\Lambda}(t)}\right)$	natural
$\mathrm{LNorm}(\mu, \beta)$	$\phi^{-1}\left(1 - e^{-\hat{\Lambda}(t)}\right)$	logarithmic
$\mathrm{Logis}(\mu, \beta)$	$\log\left(e^{\hat{\Lambda}(t)} - 1\right)$	natural
$\mathrm{LLogis}(\alpha, \beta)$	$\log\left(e^{\hat{\Lambda}(t)} - 1\right)$	logarithmic
$\mathrm{B}(\alpha, \gamma)$	$F^{-1}_{\mathrm{B}(\alpha,\gamma)}\left(1 - e^{-\hat{\Lambda}(t)}\right)$	natural
$\mathrm{ExpPow}(\alpha, \beta)$	$\log\left(\log\left(\hat{\Lambda}(t) + 1\right)\right)$	logarithmic
$\mathrm{ExpWei}(\alpha, \gamma, \beta)$	$-\log\left(1-\left(1- e^{-\hat{\Lambda}(t)}\right)^{\frac{1}{\gamma}}\right)$	logarithmic

Hence this transformation of the estimated cumulative hazard is plotted against t, if the scale is natural, or against $\log t$, if the scale is logarithmic. Since the cumulative hazard estimator is an step function, with jumps in those times where

an event is observed, the plotted points will only correspond to the failure times where $Y_i = T_i$. Note that the censored failure times do not appear in the plot. Their contribution to the cumulative hazard plot is not in the graphical output itself but in the estimation of the distribution parameters.

4.3. Exact Goodness of Fit test for Type I and Type II censored data – `Grane.test` function

In Grané (2012) [**Gra12**] it was presented an algorithm to compute the density function f of the goodness of fit test statistic Q_n (for simplicity, we will denote Q_n to Q_{n_I} and to the $Q_{n_{II}}$ introduced in Section 3.3 indistinctly). We implemented this algorithm in `R` with the aim we can compute

$$P(x < Q_n) = \int_{-\infty}^{Q_n} f(x)dx$$

and decide if we can accept the null hypothesis

$$H_0 : F = F_0$$

or we must reject it.

The function `Grane.test` reproduces the goodness of fit test studied in Section 3.3, computing the Q_n statistic from the given survival data and the probability $P(x<Q_n)$ to use it as a $p-value$ for the test.

4.3.1. Usage and input arguments

The usage of the `Grane.test` function is the following:

```
Grane.test (time, cens, distr, cens.type, cens.time,
            beta.limits = c(0,1), Q.plot = "TRUE",
            parameters = list(shape = NULL, shape2 = NULL,
                              location = NULL, scale = NULL))
```

The `Grane.test` function has seven input arguments, which correspond to

`time`	The vector of times until the studied event
`cens`	The vector indicating the censored observations

`distr`	A string specifying the distribution to be tested. The possible distributions are the weibull (`"weibull"`), the gumbel (`"gumbel"`), the normal (`"norm"`), the log-normal (`"lnorm"`), the logistic (`"logis"`), the log-logistic (`"loglogis"`), the beta (`"beta"`), the exponential power (`"exppower"`) and the exponentiated weibull (`"expweibull"`).
`cens.type`	Censoring type. This argument can take the values `"I"` (when data is right-censored of Type I) and `"II"` (when data is right-censored of Type II).
`cens.time`	A number specifying the censoring preset time when data is Type I censored. This argument is only required when `cens.type="I"`.
`beta.limits`	A two components vector corresponding to the lower and upper bounds of the Beta distribution. This argument is only required if the beta distribution is considered. By default, it is set to `c(0,1)`.
`Q.plot`	A logical value indicating if a additional plot of the density function must be plot or not. By default it is set to `"TRUE"`.
`parameters`	A list specifying the parameters of the theoretical distribution. By default they are set to `NULL` and estimated with the maximum likelihood estimate. This argument is only considered if all parameters of the tested distribution are specified.

4.3.2. Output

The function returns a list containing two vectors. The first vector, called `test`, contains information about the goodness of fit test result. Namely the components of the `test` vector are the value of the Q_n statistic, the value of $P(x < Q)$ and the absolute error of this computed probability. The second vector is called `param` and provides the values of the parameters of the tested distribution. If the user has set the parameters manually (introducing them to the function via the `parameters` argument), these values will be the ones included in the output, otherwise the provided values of the parameters will be the estimates by maximum likelihood estimation.

For example, if we simulated a type II right-censored data set of sample size 10 from a Normal(3,1) with only 6 events observed and we applied the `Grane.test` function to asses the goodness of fit of a Weibull distribution, the output of the calling `Grane.test(time,cens,"weibull", cens.type="II")` will be

```
$test
   Q.stat    p.value     abs.error
0.8718812  0.3324608  3.691057e-15
```

```
$param
   shape    scale
2.261221 3.904680
```

Optionally, an addition plot can be included in the output. This plot shows the density function and paints in red the area under the curve for times smaller than Q_n. In Figure 4.5 one can see this additional plot corresponding to the previous example.

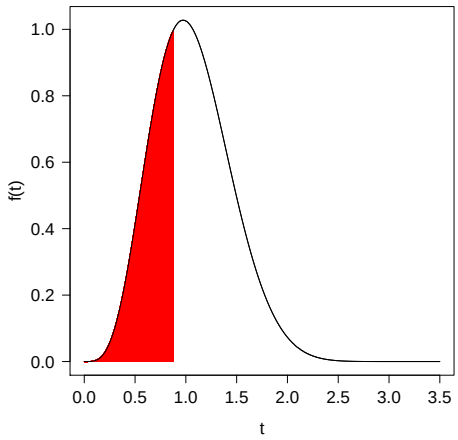


FIG. 4.5. Additional plot of the `Grane.test` function.

Note that the plot does not provide extra information to decide if the distribution is appropriate or not, but since we had some problems in computing the density function f (these problems are explained in Section 4.3.4), we decided to add this plot as a graphical check that the function f has been correctly computed.

4.3.3. How does it work?

The `Grane.test` function mainly follows the steps proposed in Grané (2012) [**Gra12**]. Next we present the details of how these steps has been implemented.

Estimation of distribution parameters

Except in the case that the user has specified the distribution parameters, we will estimate them by maximum likelihood using the `fitdistcens` function.

Computation of the Q_n statistic

If (t_i, δ_i), $1 \leq i \leq n$ is the introduced survival data of sample size n, we compute $x_i = F_0(t_i)$, where F_0 is the considered theoretical distribution with the specified parameters or the estimated ones by maximum likelihood, for each t_i such that $\delta_i = 1$. If data suffer right censoring of Type I, $x_{r+1} = F_0(y^*)$ is also computed where y^* is the censoring time and r the number of observed events.

Then we compute the a_i coefficients as shown in Proposition 3.2. To do that we define $\tilde{r}$ as $r+1$ if data are right-censored of Type I, or as r if data are right-censored of Type II. Note that $\tilde{r}$ is the number of available x_i. So a_i, for $1 \leq i \leq \tilde{r}$, is computed as

$$a_i = \frac{6((2i-1)\tilde{r} - n^2)}{n^2\tilde{r}} \text{ for } 1 \leq i \leq \tilde{r}-1 \text{ and } a_{\tilde{r}} = \frac{6(\tilde{r}-1)(n^2 - \tilde{r}(\tilde{r}-1))}{n^2\tilde{r}}.$$

Finally the Q_n statistic is computed as

$$Q_n = \sum_{i=1}^{\tilde{r}} a_i x_i.$$

Computation of the density f of Q_n

To compute the density function f of the Q_n statistic we mainly follow the algorithm proposed by Dwass (1961), Matsunawa (1985) and Ramallingam (1989) explained in Section 3.3.

Computation of the $p-value$

The $p-value$ of the Q_n statistic is defined as

$$p-value = P(T < Q_n) = \int_{-\infty}^{Q_n} f(t)\,dt.$$

We used the `integrate` function to compute the integration of the density function f until the Q_n value.

4.3.4. Limitations

While implementing this method to `R` we came across with some limitations.

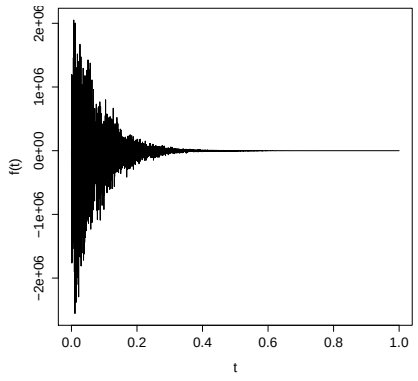
The first one was to compute the j-th derivative of function G_l (defined in page 35) when computing the density function f of the statistic Q_n. Remember that j goes from 0 to $\nu_l - 1$, where ν_l corresponds to the multiplicity of the coefficient b_l. The problem is we do not know how to compute j-th derivatives for $j \geq 2$ using `R`. We studied the multiplicities of the b_l coefficients and we saw that for $n < 1000$ these multiplicities are not greater than two. Hence we implemented the method considering that coefficients b_l can have at most multiplicity two and we used the `jacobian` function of the `numDeriv` package to compute the first derivative of function G_l when it is needed.

The second limitation, and the one with the greater impact, is that the algorithm is based on polynomials of degree n with all their roots concentrated in an small interval and its derivatives, so the precision needed to do the calculus properly is very high. When n increases, there is a enormous loss of precision and we failed to compute the density function properly. In Figure 4.6 there are shown some graphical examples of the computed density function. Note that only the density function plotted in Figure 4.6C is consistent with the definition of a density function, since densities depicted in Figures 4.6A and 4.6B present negative values. In fact,

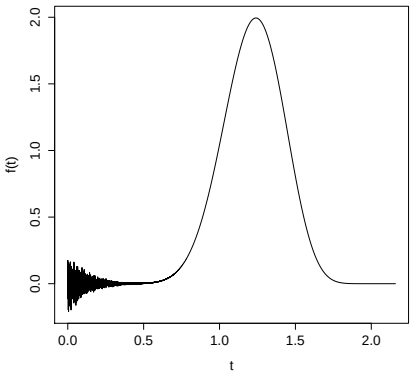
these parts of the graphics that seem to be drawn by a seismograph appears due to the lack of precision.

In some cases the seismograph drawn part appears only in a tail of the density distribution and it is clear that the real value in this part should be zero. This is the case of Figure 4.6B. We managed to modify the f function to set to zero these tails and getting a consistent density function. However we could not fix the problem when the computed density resembles the one shown in 4.6A.

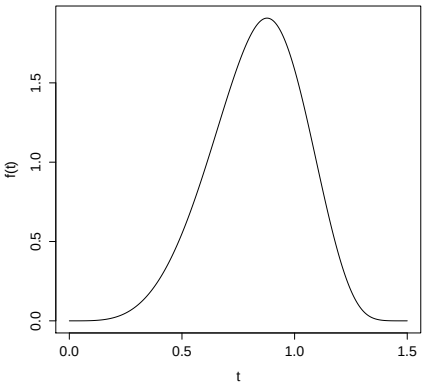
After all the modifications and adaptations, the implemented method only works for certain sample sizes. It works for sample sizes less than 20 and also for sample sizes n where $20 \leq n \leq 30$ and $r \leq \lfloor \frac{n}{2} \rfloor + 4$. These sample sizes are very small and the method tends to always accept the null hypothesis.



(A) Example of failing to compute f



(B) Another example of failing to compute f



(C) Example of success in computing f

FIG. 4.6. Graphical representation of the computed density function f.

4.4. Kolomorov-Smirnov test for right-censored data – KScens function

Fleming et al. (1980) proposed a modified Kolmogorov-Smirnov test to use with right-censored data in their paper "Modified Kolmogorov-Smirnov test procedure with application to arbitrarily right-censored data". The `KScens` reproduces this test and given survival data and a theoretical distribution, the function returns the needed information to decide if the theoretical distribution is appropriate for the data or not.

4.4.1. Usage and input arguments

The usage of the `KScens` function is the following:

```
KScens (x, c, distr,
        beta.limits = c(0,1),
        parameters = list(shape = NULL, shape2 = NULL,
                          location = NULL, scale = NULL))
```

Note that the function has five input arguments, which correspond to

`x`	The vector of times until the studied event
`c`	The vector indicating the censored observations
`distr`	A string specifying the distribution to be tested. The possible distributions are the weibull (`"weibull"`), the gumbel (`"gumbel"`), the normal (`"norm"`), the log-normal (`"lnorm"`), the logistic (`"logis"`), the log-logistic (`"loglogis"`), the beta (`"beta"`), the exponential power (`"exppower"`) and the exponentiated weibull (`"expweibull"`).
`beta.limits`	A two components vector corresponding to the lower and upper bounds of the Beta distribution. This argument is only required if the beta distribution is considered. By default, it is set to `c(0,1)`.
`parameters`	A list specifying the parameters of the theoretical distribution. By default they are set to `NULL` and estimated with the maximum likelihood estimate. This argument is only considered if all parameters of the tested distribution are specified.

4.4.2. Output

The output of the `KScens` function is a list with three vectors. The first one is called `test` and contains the estimated $p - value$ (`p.value`) , the value of the modified Kolmogorov-Smirnov statistic (`A`), the estimation of the image of the last

recorded time y_m under the null hypothesis (`F(ym)`) defined as

$$1 - \frac{1}{2}\left(\hat{S}_n(y_m) + S_0(y_m)\right),$$

and the last recorded time y_m (`ym`).

The second vector is called `distr` and reminds the user which distribution is being fitted, and the third vector of the list, named `param`, contains the values of the distribution parameters (the values introduced by the user or the estimated ones if the parameter values had not been specified).

To illustrate the output of the `KScens` function, we simulated a 250 sample sized right-censored data set from a standard Lognormal distribution with nearly a 30% of censored observations. Let us suppose that we want to test the goodness of fit of the Normal distribution to these simulated data using the `KScens` function, hence we type the sentence

```
KScens(x,c,"norm")
```

in the `R` console, being `x` the vector of times and `c` the vector of the censoring indicators, getting the following output:

```
$test
     p.value           A     F(y_m)  last.time
3.468839e-09   3.175893  0.9729548    5.616831

$distr
[1] "norm"

$param
location    scale
1.357758 1.090613
```

The $p - value$ provided is extremely small, so the Normal distribution is not an appropriate distribution for the data.

4.4.3. How does it work?

At the end of the Fleming et al. paper [**FOOH80**] the authors proposed a procedure to calculate the generalized two-sided Kolmogorov-Smirnov goodness of fit test based on upon the statistic $\sqrt{n}\tilde{D}_n(y_m)$. The `KScens` function mainly follows the steps of the proposed procedure and next the details of how the different steps are carried out will be provided.

Estimation of distribution parameters

If the user had not specified the values of the distribution parameters, these are estimate by maximum likelihood. To compute this estimations we used the function `fitdistcens`.

Estimation of the Survival function for observed times

A survival object is created via the `R` function `Surv` from the failure times and the censoring indicators vector. This survival object is used to compute the Nelson-Aalen estimator for the cumulative hazard function, $\hat{\Lambda}_{NA}$. By means of the `survfit` function, the derived survival estimator is taken as an argument and the Nelson-Aalen estimator is computed taking in account that we want to break the ties and assume that deaths occur infinitesimally earlier than censoring. Then the survival is derived from the equation

$$\hat{S}_{NA}(t) = e^{-\hat{\Lambda}_{NA}}.$$

Estimation of the Survival function for censored times

To estimate the survival function for censored times, it is done the same as for estimating the survival for observed times but inverting the censoring indicator. In this case the vector used as censoring vector values 1 when the observation is censored and 0 otherwise.

In order to break the ties between observed and censored times and to force that observed times happen infinitesimally earlier than censored times, we add a number smaller than 10^{-10} to the censored observations tied with non-censored observations.

Computation of the Kolmogorov-Smirnov statistic

Adopting the notation introduced in Section 3.4, we compute the value of the Kolmogorov-Smirnov statistic A as follows:

$$A = \max\left\{|Y_n(t_i^-)|, |Y_n(t_i)|, |Y_n(y_m)| : i = 1, \ldots, d\right\},$$

where $Y_n(t_i^-)$ and $Y_n(t_i)$ are computed following the steps (ii) and (iii) explained at the end of Section 3.4.

Estimation of the $p - value$

Firstly the $p(x, y)$ function is defined as in Lemma 3.4, and then the $p - value$ is estimated by $2p(A, R)$ when $p(A, R) < 0.40$, being R the estimated image of y_m under the null hypothesis computed as

$$R = 1 - \frac{1}{2}\left(\hat{S}_n(y_m) + S_0(y_m)\right).$$

If $p(A, R) > 0.40$, since the above approximation is not longer valid, the $p - value$ is set to `NULL` and the message

```
Warning! The p-value can not be estimated because p(A,F(ym))>0.4
```

is shown on the screen.

4.4.4. Limitations

The limitations of the `KScens` function are not computational but that the provided estimation of the $p - value$ is only valid under certain condition, which is

$p(A,R) < 0.4$. When the condition to estimate the $p-value$ as $2p(A,R)$ is not fulfilled, the `KScens` function does not give any estimation for the $p-value$ but provide the value of the statistic (`A`) and the estimation of $F(y_m)$ (`F(ym)`).

Koziol and Byar (1975) [**KB75**] provided the tabulated critical values for this test (see Table 3.7), so with the `A` and `F(ym)` quantities the user can go to this table (we also added it in `R` under the name `KS.table`) and estimate the $p-value$. For example, suppose that we want to assess the goodness of fit of the Lognormal distribution to the Lognormal simulated data used before. In this case the output of the `KScens` function is

```
 Warning! The p-value can not be estimated because p(A,F(y_m))>0.4

$test
        A    F(y_m) last.time
0.5238927 0.9505914 5.6168310

$distr
[1] "lnorm"

$param
  location      scale
-0.0288776  1.0334896
```

The function could not provide us a $p-value$ but if we use the Koziol and Byar table (see Figure 4.7) we found that the $p-value$ is approximately 0.95, suggesting us that the Lognormal distribution fits well to the data.

	$F(y_m)$									
α	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90	1.00
0.990	0.1587	0.2232	0.2717	0.3115	0.3454	0.3747	0.3999	0.4209	0.4362	0.4410
0.975	0.1761	0.2473	0.3006	0.3441	0.3810	0.4125	0.4394	0.4612	0.4764	0.4806
0.950	0.1938	0.2718	0.3299	0.3771	0.4168	0.4504	0.4786	0.5011	0.5160	0.5196
0.900	0.2182	0.3054	0.3700	0.4219	0.4652	0.5014	0.5311	0.5540	0.5683	0.5712
0.850	0.2376	0.3321	0.4015	0.4571	0.5029	0.5409	0.5716	0.5946	0.6082	0.6106
0.800	0.2550	0.3559	0.4297	0.4883	0.5363	0.5756	0.6069	0.6300	0.6428	0.6448
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

FIG. 4.7. Example of how to use the Koziol and Byar table of Kolmogorov-Smirnov critical values for right-censored data.

Chapter 5
Discussion and further research

In this master's degree thesis we presented four different methods to assess goodness of fit for parametric survival models with right-censored data. Since there does not exists much references about this type of goodness of fit tests for right-censored data, this work can be seen as a little guide of how to assess if a distribution is appropriate for the data when these present right-censored observations.

A part from explain, theoretically, how these goodness of fit methods work, they were also implemented in `R` creating a local library with functions that aid the user to assess the goodness of fit of a distribution to right-censored data. Two functions were built as implementation of the graphical methods presented. The `prob.plots` function one, based on probability plots (including the P-P plot, the Q-Q plot, the stabilised probability plot and the rescaled plot), mainly shows if a theoretical distribution fits well to the right-censored data or not; and the `CumHazPlot` function, which is based on cumulative hazard plots, has been constructed as a tool to compare the goodness of fit of distinct distributions to the same data.

The analytical goodness of fit tests explained in this work were also implemented in `R` via the functions `Grane.test` and `KScens`, but they have some limitation. The `Grane.test` function can only be used when we are dealing with type I or type II right-censored data. Moreover, the computations involved in this test need a very high precision and our `R` implementation does not reach it. Due to this lack of precision, the `Grane.test` function presented only works properly for data sets of sample sizes up to twenty and also for sample sizes n where $20 \leq n \leq 30$ and $r \leq \lfloor \frac{n}{2} \rfloor + 4$, where r is the number of observed events. Another inconvenient of this function is that since it only can be applied to small sample sizes, it tends to always accept the null hypothesis. As a further work, a reformulation of the computations and a better implementation of them can be study with the aim to avoid the lack of precision problem and improve the `Grane.test` functionality. The `KScens` function can be applied to any type of right censoring pattern, as long as there exists independence between the time and the censoring. However it also presents a limitation since the estimated $p-value$ that the `KScens` function returns is only valid under certain conditions. With the goal to overcome this inconvenient we load in the library a table with the tabulated $p-values$ for the Kolmogorov-Smirnov goodness of fit test for right-censored data. Therefore, if the `KScens` function can

provide us an estimation of the $p - value$, we can resort to the table to find the corresponding $p - value$.

As a further work, a similar study like the one done in this work can be carried out when considering interval-censored data. For example, most of the graphical methods to assess goodness of fit presented in this work (the P-P plot, the Q-Q plot, the stabilised probability plot and the cumulative hazard plot) were initially developed for uncensored data, but only modifying the way that the parameters are estimated that these plots can also be applied to right-censored data. Unfortunately, since when data is interval-censored we do not the exact time of failure but an interval where the failure has taken place, the plots presented in this work are not longer applicable. However one can investigate if some modification of these plots, for example using Turnbull intervals, have been studied to be used as graphical methods to assess goodness of fit when data are interval-censored.

In Grané (2012) [**Gra12**], apart from the goodness of fit test for type I and type II right-censored data, a goodness of fit test for interval-censored data has been also introduced. Therefore, one can search if modifications of the Kolmogorov-Smirnov or of the Cramér-vonMises tests have been proposed to be used with interval-censored data and the existence of other goodness of fit tests may be investigated.

References

Aal76. O Aalen, *Nonparametric inference in connection with multiple decrement models*, Scandinavian Journal of Statistics **3** (1976), 15 – 27.

Aal78. O. O. Aalen, *Nonparametric inference for a family of counting processes*, Annals of Statistics **6** (1978), 701–726.

AD52. T. W. Anderson and D. A. Darling, *Asymptotic theory of certain "goodness-of-fit" criteria based on stochastic processes*, Annals of Mathematical Statistics **23** (1952), 193 – 212.

BD73. D. R. Barr and T. Davidson, *A kolmogorov-smirnov test for censored samples*, Technometrics **15** (1973), no. 4, 739 – 757.

Bil68. P Billingsley, *Convergence of probability measure*, Wiley Series in probability and Mathematical Statistics: Tracts on probability and statistics, Wiley, 1968.

Can33. F. P. Cantelli, *Sulla determinazione empirica della leggi di probabilità*, G. Inst. Ital. Attuari **4** (1933), 221 – 424.

CO84. D. R. Cox and D. Oakes, *Analysis of survival data*, Chapman & Hall/CRC Monographs on Statistics & Applied Probability, Taylor & Francis, 1984.

Cra28. H. Cramér, *On the composition of elementary errors*, Scandinavian Actuarial Journal **1928** (1928), no. 1, 13 – 74.

DG08. P. Delicado and M. N. Goria, *A small sample comparison of maximum likelihood, moments and l-moments methods for the asymmetric exponential power distribution*, Comput. Stat. Data Anal. **52** (2008), no. 3, 1661–1673.

DM78. R. Dufour and J. R. Maag, *Distribution results for modified kolmogorov-smirnov statistics for truncated or censored samples*, Technometrics **20** (1978), 29 – 32.

DMD14. M. L. Delignette-Muller and C. Dutang, *fitdistrplus: Help to fit of a parametric distribution to non-censored or censored data*, 2014, R package version 1.0-2.

DS86. R. B. D'Agostino and M. A. Stephens (eds.), *Goodness-of-fit techniques*, Statistics, textbooks and monographs, New York. Marcel Dekker, Inc., 1986.

Dwa61. M. Dwass, *The distribution of linear combinations of random divisions of an interval*, Trabajos de Estadística e Investigación Operativa **12** (1961), 11–17.

FG03. J. Fortiana and A. Grané, *Goodness-of-fit tests based on maximum correlations and their orthogonal decompositions*, Journal of the Royal Statistical Society: Series B (Statistical Methodology) **65** (2003), no. 1, 115–126.

FH81. T. R. Fleming and D. P. Harrington, *A class of hypothesis test for one and two sample censored survival data*, Communications in Statistics – Theory and methods **10** (1981), no. 8, 763 – 794.

FOOH80. T. R. Fleming, J. R. O'Fallon, P. C. O'Brien, and D. P. Harrington, *Modified kolmogorov-smirnov test procedure with application to arbitrarily right-censored data*, Biometrics **36** (1980), 607 – 625.

Gli33. V. Glivenko, *Sulla determinazione empirica di probabilità*, G. Inst. Ital. Attuari **4** (1933), 92 – 99.

Gra12. A. Grané, *Exact goodness-of-fit tests for censored data*, Annals of the Institute of Statistical Mathematics **64** (2012), no. 6, 1187–1203.

Hjo90. N. L. Hjort, *Goodness of fit tests in models for life history data based on cumulative hazard rates*, Annals of Statistics **18** (1990), no. 3, 1221 – 1258.

HT86. M. G. Habib and D. R. Thomas, *Chi-square goodness-of-fit test of fit randomly censored data*, Annals of Statistics **14** (1986), no. 2, 759 – 765.

KB75. J. R. Koziol and D. P. Byar, *Percentage points of the asymptotic distributions of one and two sample k-s statistics for truncated or censored data*, Technometrics **17** (1975), 507 – 510.

KG76. J. A. Koziol and S. B. Green, *A cramér-von mises statistic for randomly censored data*, Biometrika **63** (1976), no. 3, 465 – 475.

Kim93. J. H. Kim, *Chi-square goodness-of-fit test of fit randomly censored data*, Annals of Statistics **21** (1993), no. 3, 1621 – 1639.

KM55. E. L. Kaplan and P. Meier, *Nonparametric estimation from incomplete observations*, Journal of the American Statistical Association **53** (1955), 457–481.

KM03. J. P. Klein and M. L. Moeschberger, *Survival analysis: Techniques for censored and truncated data*, Statistics for Biology and Health, Springer, 2003.

Kol33. A. Kolmogorov, *Sulla determinazione empirica di una legge di distribuzione*, G. Ist. Ital. Attuari **4** (1933), 83 – 91.

Mat85. T. Matsunawa, *The exact and approximate distribution of linear combinations of selected order statistics from a uniform distribution*, Annals of the Institute of Statistical Mathematics **37** (1985), 1–16.

Mic83. J. R. Michael, *The stabilized probability plot*, Biometrika **70** (1983), no. 1, 11–17.

MM80. D. P. Mihalko and D. S. Moore, *Chi-square test of fit for type ii censored data*, Annals of Statistics **8** (1980), no. 3, 625 – 644.

Nel72. W. Nelson, *Theory and applications of hazard plotting for censored failure data*, Technometrics **14** (1972), no. 4, 945–965.

PS76. A. N. Pettitt and M.A. Stephens, *Modified cramér-von mises statistics for censored data*, Biometrika **63** (1976), no. 2, 291 – 298.

Ram89. T. Ramallingam, *Symbolic computing the exact distributions of l-statistics from a uniform distribution*, Annals of the Institute of Statistical Mathematics **41** (1989), 677 – 681.

SB75. R. M. Smith and L. J. Bain, *An exponential power life-testing distribution*, Communications in Statistics - Theory and Methods **4** (1975), 469–481.

Sch81. H. M. Schey, *The asymptotic distribution of the one-sided kolmogorov-smirnov statistic for truncated data*, Communications in Statistics – Theory and methods **6** (1981), 1361 – 1365.

Smi39. N. V. Smirnov, *On the estimation of the discrepancy between empirical curves of distribution for two independent samples*, Bulletin Mathematique de l'Université de Moscou **2** (1939), no. 2, 3 – 14.

Smi48. ———, *Table for estimating the goodness of fit of empirical distributions*, Annals of Mathematical Statistics **19** (1948), 279 – 281.

TW78. B. W. Turnbull and L. Weiss, *A likelihood ratio statistic for testing goodness of fit with randomly censored data*, Biometrika **34** (1978), 367 – 375.

vM28. R. von Mises, *Wahrscheinlichkeit, statistik und wahrheit. wien 1928. iv, 189 s.*, Schriften zur wissenschaftlichen Weltauffassung, J. Springer, 1928.

Wei51. W Weibull, *A statistical distribution function of wide applicability*, Journal of Applied Mechanics **18** (1951), 293 – 297.

WT92. L. A. Waller and B. W. Turnbull, *Probability plotting with censored data*, The American Statistician **46** (1992), no. 1, 5–12.

Appendix A
prob.plots code

```
prob.plots <- function(time,
                       cens,
                       distribution,
                       beta.limits=c(0,1),
                       plots = c("PP","QQ","SP","ER"),
                       colour = c("green4","deepskyblue4",
                                  "yellow3","mediumvioletred"),
                       parameters = list(shape = NULL, shape2 = NULL,
                                         location = NULL, scale = NULL)
){
  # Load the required packages
  require(survival)
  require(fitdistrplus)

  # Tranform the input data to the needed format
  n <- length(time)
  survKM <- survfit(Surv(time, cens)~1, type='kaplan-meier')
  data <- data.frame(left=time,right=ifelse(cens==1,time,NA))

  # Compute the event times
  t<-summary(survKM)$time

  # Compute the survival at the event times
  surv.value<-summary(survKM)$surv

  uncensored<-rep(1,length(t))
  u.point.surv<-survfit(Surv(t,uncensored)~1, type='kaplan-meier')
  u.point<-1-u.point.surv$surv

  empirical_f <- rbind(c(0,t,Inf), c(0,u.point,1))
  u.estimate <- rep(0,length(u.point))

  in.p <- parameters
```

```
  # Probability plots
  # exponential
  if(distribution=="exp"){
    if(is.null(in.p$scale)){
      fit.exp <- fitdistcens(data,"exp")
      rate.exp <- unname(fit.exp$estimate[1])
    }
    else rate.exp <- 1/in.p$scale
    theor.PP <- pexp(t, rate.exp)
    theor.QQ <- qexp(1-surv.value, rate.exp)
    out.p <- list(distr = "Exponential", scale = 1/rate.exp)
  }

  # weibull
  if(distribution=="weibull"){
    if(is.null(in.p$shape) || is.null(in.p$scale)){
      fit.wei <- fitdistcens(data,"weibull")
      shape.wei <- unname(fit.wei$estimate[1])
      scale.wei <- unname(fit.wei$estimate[2])
    }
    else{
      shape.wei <- in.p$shape
      scale.wei <- in.p$scale
    }
    theor.PP <- pweibull(t, shape.wei, scale.wei)
    theor.QQ <- qweibull(1-surv.value, shape.wei, scale.wei)
    out.p <- list(distr = "Weibull", shape = shape.wei, scale = scale.wei)
  }

  # log-weibull (gumbel)
  if(distribution=="gumbel"){
    dgumbel <<- function(x,mu,beta){
      1/beta*exp((x-mu)/beta)*exp(-exp((x-mu)/beta))}
    pgumbel <<- function(q,mu,beta) 1-exp(-exp((q-mu)/beta))
    qgumbel <<- function(p,mu,beta) log(log(1/(1-p)))*beta+mu
    if(is.null(in.p$location) || is.null(in.p$scale)){
      fit.gum <- fitdistcens(data,"gumbel",start=list(mu=-3,beta=3))
      loc.gum <- unname(fit.gum$estimate[1])
      scale.gum <- unname(fit.gum$estimate[2])
    }
    else{
      loc.gum <- in.p$location
      scale.gum <- in.p$scale
    }
    theor.PP <- pgumbel(t, loc.gum, scale.gum)
    theor.QQ <- qgumbel(1-surv.value, loc.gum, scale.gum)
    out.p <- list(distr = "Gumbel", location = loc.gum, scale = scale.gum)
  }
```

```
# normal
if(distribution=="norm"){
  if(is.null(in.p$location) || is.null(in.p$scale)){
    fit.norm <- fitdistcens(data,"norm")
    loc.norm <- unname(fit.norm$estimate[1])
    scale.norm <- unname(fit.norm$estimate[2])
  }
  else{
    loc.norm <- in.p$location
    scale.norm <- in.p$scale
  }
  theor.PP <- pnorm(t, loc.norm, scale.norm)
  theor.QQ <- qnorm(1-surv.value, loc.norm, scale.norm)
  out.p <- list(distr = "Normal", location = loc.norm, scale = scale.norm)
}

# log-normal
if(distribution=="lnorm"){
  if(is.null(in.p$location) || is.null(in.p$scale)){
    fit.lnorm <- fitdistcens(data, "lnorm")
    loc.lnorm <- unname(fit.lnorm$estimate[1])
    scale.lnorm <- unname(fit.lnorm$estimate[2])
  }
  else{
    loc.lnorm <- in.p$location
    scale.lnorm <- in.p$scale
  }
  theor.PP <- plnorm(t, loc.lnorm, scale.lnorm)
  theor.QQ <- qlnorm(1-surv.value, loc.lnorm, scale.lnorm)
  out.p <- list(distr = "Log-normal", location = loc.lnorm, scale = scale.lnorm)
}

# logística
if(distribution=="logis"){
  if(is.null(in.p$location) || is.null(in.p$scale)){
    fit.log <- fitdistcens(data,"logis")
    loc.logis <- unname(fit.log$estimate[1])
    scale.logis <- unname(fit.log$estimate[2])
  }
  else{
    loc.logis <- in.p$location
    scale.logis <- in.p$scale
  }
  theor.PP <- plogis(t, loc.logis, scale.logis)
  theor.QQ <- qlogis(1-surv.value, loc.logis, scale.logis)
  out.p <- list(distr = "Logistic", location = loc.logis, scale = scale.logis)
}
```

```
  # log-logística
  if(distribution=="loglogis"){
    dloglogis <<- function(x,alpha,beta) {
      (alpha*beta^(-alpha)*x^(alpha-1))/(1+(x/beta)^alpha)^2}
    ploglogis <<- function(q,alpha,beta) 1/(1+(q/beta)^(-alpha))
    qloglogis <<- function(p,alpha,beta) beta*(p/(1-p))^(1/alpha)
    if(is.null(in.p$shape) || is.null(in.p$scale)){
      fit.loglog <- fitdistcens(data,"loglogis", start=list(alpha=1,beta=1))
      shape.loglogis <- unname(fit.loglog$estimate[1])
      scale.loglogis <- unname(fit.loglog$estimate[2])
    }
    else{
      shape.loglogis <- in.p$shape
      scale.loglogis <- in.p$scale
    }
    theor.PP <- ploglogis(t, shape.loglogis, scale.loglogis)
    theor.QQ <- qloglogis(1-surv.value, shape.loglogis, scale.loglogis)
    out.p <- list(distr = "Log-logistic", shape = shape.loglogis,
                  scale = scale.loglogis)
  }

  # beta
  if(distribution=="beta"){
    a.beta<-beta.limits[1]
    b.beta<-beta.limits[2]
    if(is.null(in.p$shape) || is.null(in.p$shape2)){
      fit.beta <- fitdistcens((data-a.beta)/(b.beta-a.beta),"beta")
      shape1.beta <- unname(fit.beta$estimate[1])
      shape2.beta <- unname(fit.beta$estimate[2])
    }
    else{
      shape1.beta <- in.p$shape
      shape2.beta <- in.p$shape2
    }
    theor.PP <- pbeta((t-a.beta)/(b.beta-a.beta), shape1.beta, shape2.beta)
    theor.QQ <- qbeta((1-surv.value), shape1.beta, shape2.beta)*
      (b.beta-a.beta)+a.beta
    out.p <- list(distr = "Beta", shape1 = shape1.beta, shape2 = shape2.beta,
                  interval.domain = beta.limits)
  }

  # Exponentiated Weibull
  if(distribution=="expweibull"){
    dexpwei <<- function(x,alpha,gamma,beta){
      gamma*alpha*beta^alpha*x^(alpha-1)*
        exp(-(beta*x)^alpha)*(1-exp(-(beta*x)^alpha))^(gamma-1)}
    pexpwei <<- function(q,alpha,gamma,beta) (1-exp(-(beta*q)^alpha))^gamma
```

```
  qexpwei <<- function(p,alpha,gamma,beta){
    (log(1/(1-p^(1/gamma))))^(1/alpha)/beta}
  if(is.null(in.p$shape) || is.null(in.p$shape2) || is.null(in.p$scale)){
    fit.expwei <- fitdistcens(data,"expwei",
                              start=list(alpha=1,gamma=1,beta=1))
    shape1.expwei <- unname(fit.expwei$estimate[1])
    shape2.expwei <- unname(fit.expwei$estimate[2])
    scale.expwei <- unname(fit.expwei$estimate[3])
  }
  else{
    shape1.expwei <- in.p$shape
    shape2.expwei <- in.p$shape2
    scale.expwei <- in.p$scale
  }
  theor.PP <- pexpwei(t, shape1.expwei, shape2.expwei, scale.expwei)
  theor.QQ <- qexpwei(1-surv.value, shape1.expwei,
                      shape2.expwei, scale.expwei)
  out.p <- list(distr = "Exponetiated Weibull", shape1 = shape1.expwei,
                shape2 = shape2.expwei, scale = scale.expwei)
}

# Exponential power
if(distribution=="exppower"){
  dexppow <<- function(x,alpha,beta){
    alpha*beta^alpha*x^(alpha-1)*
      exp((beta*x)^alpha)*exp(1-exp((beta*x)^alpha))}
  pexppow <<- function(q,alpha,beta) 1-exp(1-exp((beta*q)^alpha))
  qexppow <<- function(p,alpha,beta) (log(1-log(1-p)))^(1/alpha)/beta
  if(is.null(in.p$shape) || is.null(in.p$scale)){
    fit.exppow <- fitdistcens(data,"exppow",start=list(alpha=0.5,beta=0.5))
    shape.exppow <- unname(fit.exppow$estimate[1])
    scale.exppow <- unname(fit.exppow$estimate[2])
  }
  else{
    shape.exppow <- in.p$shape
    scale.exppow <- in.p$scale
  }
  theor.PP <- pexppow(t, shape.exppow, scale.exppow)
  theor.QQ <- qexppow(1-surv.value, shape.exppow, scale.exppow)
  out.p <- list(distr = "Exponential Power", shape = shape.exppow,
                scale = scale.exppow)
}

index <- 0
for (i in 1:length(u.point)){
  while (theor.QQ[i]>empirical_f[1,index+1]) index <- index + 1
  if(index!=0) u.estimate[i] <- empirical_f[2,index]
}
```

```
  n.col <- length(colour)
  howmany <- length(plots)
  if(howmany==1) {m<-matrix(c(1), nrow = 1, ncol = 1)}
  else if(howmany==2) {m<-matrix(c(1, 2), nrow = 1, ncol = 2)}
  else if(howmany==3) {m<-matrix(c(1, 0, 1,  3, 2, 3, 2, 0),
                                 nrow = 2, ncol = 4)}
  else{m<-matrix(c(1, 3, 2, 4), nrow = 2, ncol = 2)}
  layout(m)
  par(col=1, las=1, mar=c(4.5,5,2,1))

  for (i in 1:howmany){
    if(plots[i]=="PP"){
      plot(1-surv.value, theor.PP, col = colour[0%%n.col+1],
           xlab=expression(hat(F)(t)), ylab=expression(hat(F)[0](t)),
           main="PP plot")
      lines(c(0,1),c(0,1), type = 'l')
    }
    if(plots[i]=="QQ"){
      plot(t, theor.QQ, col = colour[1%%n.col+1],
           xlab=expression(t),
           ylab=expression(paste(hat(F)[0]^{-1})( hat(F)(t)) ),
           main="QQ plot")
      lines(c(min(t),max(t)),c(min(t),max(t)), type = 'l')
    }
    if(plots[i]=="SP"){
      plot(2/pi*asin(sqrt(1-surv.value)), 2/pi*asin(sqrt(theor.PP)),
           col = colour[2%%n.col+1],
           xlab=expression(paste(2/pi %*% arcsin(hat(F)(t)^{1/2}))),
           ylab=expression(paste(2/pi %*% arcsin(hat(F)[0](t)^{1/2}))),
           main="SP plot")
      lines(c(0,1), c(0,1), type = 'l')
    }
    if(plots[i]=="ER"){
      plot(u.point,u.estimate,col=colour[3%%n.col+1],
           xlab=expression(hat(F)[u](t)),
           ylab=expression(hat(F)[u](paste(hat(F)[0]^{-1})(hat(F)(t)))),
           main="ER plot")
      lines(c(0,1),c(0,1),type = 'l')
    }
  }
  options(digits=7)
  out.p
}
```

Appendix B
CumHazPlot code

```
CumHazPlot<-function(time, cens,
                     distributions=c("gumbel","norm","logis",
                                     "weibull","lnorm","loglogis"),
                     beta.limits=c(0,1),
                     colour = c("orangered","darkolivegreen3","cadetblue2",
                                "red3","green4","deepskyblue4",
                                "hotpink","yellow3","mediumvioletred")
){
  # Load the required packages
  require(survival)
  require(fitdistrplus)

  # Tranform the input data to the needed format
  n <- length(time)
  data<-data.frame(left=time, right=ifelse(cens==1,time,NA))
  survNA <- survfit(Surv(time, cens)~1, type='fleming')

  # Compute the Cumulative Hazard
  Haz<-round(with(summary(survNA), -log(surv)), 6)

  # Compute the event times
  t<-summary(survNA)$time
  # Compute the survival at the event times
  surv.value<-summary(survNA)$surv

  # Setting the graphical options
  n.col <- length(colour)
  ord<--1
  if(distributions[1]=="all"){
    distributions<-c("gumbel","norm","logis",
                     "weibull","lnorm","loglogis",
                     "expweibull","beta","exppower")}
```

```
howmany=length(distributions)
if(howmany==1) {m<-matrix(c(1), nrow = 1, ncol = 1)}
else if(howmany==2) {m<-matrix(c(1, 2), nrow = 1, ncol = 2)}
else if(howmany==3) {m<-matrix(c(1, 0, 1,  3, 2, 3, 2, 0),
                               nrow = 2, ncol = 4)}
else if(howmany==4) {m<-matrix(c(1, 3, 2, 4), nrow = 2, ncol = 2)}
else if(howmany==5) {m<-matrix(c(1, 0, 1, 4, 2, 4, 2, 5, 3, 5, 3, 0),
                               nrow = 2, ncol = 6)}
else if(howmany==6) {m<-matrix(c(1, 4, 2, 5, 3, 6), nrow = 2, ncol = 3)}
else if(howmany==7) {m<-matrix(c(0,3,0,1,3,6,1,4,6,2,4,7,2,5,7,0,5,0),
                               nrow = 3, ncol = 6)}
else if(howmany==8) {m<-matrix(c(1,4,0,1,4,7,2,5,7,2,5,8,3,6,8,3,6,0),
                               nrow = 3, ncol = 6)}
else {m<-matrix(c(1,4,7,2,5,8,3,6,9), nrow = 3, ncol = 3)}

layout(m)
par(col=1, las=1, mar=c(4,4,2,1))

parameters <- list()

# Plot the desired cumulative hazard plots
for (i in 1:howmany){

  # weibull
  if(distributions[i]=="weibull"){
    ord<-ord+1
    if(min(data[,1])<0){
      plot(1,1,xlim=c(0,1),ylim=c(0,1),col="grey100", main="weibull",
           xlab="", ylab="")
      text(0.5,0.57, lab='The data is')
      text(0.5,0.43,lab='out of range')
    }
    else{
      tryCatch({
        fit.wei <- fitdistcens(data,"weibull")
        shape.wei <-fit.wei$estimate[1]
        scale.wei <- fit.wei$estimate[2]
        trans.wei <- function(Haz) log(Haz)
        parameters$weibull <- c(shape.wei, scale.wei)
        reg.wei <- function(t) shape.wei*(-log(scale.wei)+log(t))

        plot(trans.wei(Haz)~log(t),col=colour[ord%%n.col+1],
             main="weibull", ylab="")
        lines(log(t),reg.wei(t))
      }, error = function(e) e)
    }
  }
```

```
# log-weibull (gumbel)
if(distributions[i]=="gumbel"){
  ord<-ord+1
  dgumbel <<- function(x,location,scale){
                        1/scale*exp((x-location)/scale)*
                        exp(-exp((x-location)/scale))}
  pgumbel <<- function(q,location,scale){
                        1-exp(-exp((q-location)/scale))}
  tryCatch({
    fit.gum <- fitdistcens(data,"gumbel",
                          start=list(location=0,scale=2))
    shape.gum <- fit.gum$estimate[1]
    scale.gum <- fit.gum$estimate[2]
    trans.gum <- function(Haz) log(Haz)
    parameters$gumbel <- c(shape.gum, scale.gum)
    reg.gum <- function(t) (t-shape.gum)/scale.gum

    plot(trans.gum(Haz)~t,col=colour[ord%%n.col+1],
         main="gumbel", ylab="")
    lines(t,reg.gum(t))
  }, error = function(e) e)
}

# normal
if(distributions[i]=="norm"){
  ord<-ord+1
  tryCatch({
    fit.norm <- fitdistcens(data,"norm")
    loc.norm<-fit.norm$estimate[1]
    names(loc.norm) <- "location"
    scale.norm <- fit.norm$estimate[2]
    names(scale.norm) <- "scale"
    trans.norm <- function(Haz) qnorm(1-exp(-Haz))
    parameters$normal <- c(loc.norm, scale.norm)
    reg.norm <- function(t) (t-loc.norm)/scale.norm

    plot(trans.norm(Haz)~t, col=colour[ord%%n.col+1],
         main="normal", ylab="")
    lines(t,reg.norm(t))
  }, error = function(e) e)
}

# log-normal
if(distributions[i]=="lnorm"){
  ord<-ord+1
  if(min(data[,1])<0){
    plot(1,1,xlim=c(0,1),ylim=c(0,1), col="grey100",
         main="log-normal", xlab="", ylab="")
```

```
      text(0.5,0.57, lab='The data is')
      text(0.5,0.43,lab='out of range')
    }
    else{
      tryCatch({
        fit.lnorm <- fitdistcens(data, "lnorm")
        loc.lnorm <- fit.lnorm$estimate[1]
        names(loc.lnorm) <- "location"
        scale.lnorm <- fit.lnorm$estimate[2]
        names(scale.lnorm) <- "scale"
        trans.lnorm <- function(Haz) qnorm(1-exp(-Haz))
        parameters$lognormal <- c(loc.lnorm, scale.lnorm)
        reg.lnorm <- function(t) (log(t)-loc.lnorm)/scale.lnorm

        plot(trans.lnorm(Haz)~log(t),col=colour[ord%%n.col+1],
             main="log-normal", ylab="")
        lines(log(t),reg.lnorm(t))
      }, error = function(e) e)
    }
  }

  # logistic
  if(distributions[i]=="logis"){
    ord<-ord+1
    tryCatch({
      fit.log <- fitdistcens(data,"logis")
      loc.logis <- fit.log$estimate[1]
      scale.logis <- fit.log$estimate[2]
      trans.logis <- function(Haz) log(exp(Haz)-1)
      parameters$logistic <- c(loc.logis, scale.logis)
      reg.logis <- function(t) (t-loc.logis)/scale.logis

      plot(trans.logis(Haz)~t,col=colour[ord%%n.col+1],
           main="logistic", ylab="")
      lines(t,reg.logis(t))
    }, error = function(e) e)
  }

  # log-logística
  if(distributions[i]=="loglogis"){
    ord<-ord+1
    dloglogis <<- function(x,shape,scale){
                            (shape*scale^(-shape)*x^(shape-1))/
                             (1+(x/scale)^shape)^2}
    ploglogis <<- function(q,shape,scale) 1/(1+(q/scale)^(-shape))
    if(min(data[,1])<0){
      plot(1,1,xlim=c(0,1),ylim=c(0,1),col="grey100", main="log-logistic",
           xlab="", ylab="")
```

```
    text(0.5,0.57, lab='The data is')
    text(0.5,0.43,lab='out of range')
  }
  else{
    tryCatch({
      fit.loglog <- fitdistcens(data,"loglogis",
                                start=list(shape=1,scale=1))
      shape.loglogis <- fit.loglog$estimate[1]
      scale.loglogis <- fit.loglog$estimate[2]
      trans.loglogis <- function(Haz) log(exp(Haz)-1)
      parameters$loglogistic <- c(shape.loglogis, scale.loglogis)
      reg.loglogis <- function(t) shape.loglogis*(log(t)-log(scale.loglogis))

      plot(trans.loglogis(Haz)~log(t),col=colour[ord%%n.col+1],
           main="log-logistic", ylab="")
      lines(log(t),reg.loglogis(t))
    }, error = function(e) e)
  }
}

# beta
if(distributions[i]=="beta"){
  ord<-ord+1
  a.beta<-beta.limits[1]
  b.beta<-beta.limits[2]
  if(max(data[,1])>b.beta || min(data[,1])<a.beta){
    plot(1,1,xlim=c(0,1),ylim=c(0,1),col="grey100",
         main="beta", xlab="", ylab="")
    text(0.5,0.57, lab='The data is')
    text(0.5,0.43,lab='out of range')
  }
  else{
    tryCatch({
      fit.beta <- fitdistcens((data-a.beta)/(b.beta-a.beta),"beta")
      shape1.beta <- fit.beta$estimate[1]
      shape2.beta <- fit.beta$estimate[2]
      trans.beta <- function(Haz) qbeta(1-exp(-Haz),shape1.beta,shape2.beta)
      parameters$beta <- list(param = c(shape1.beta, shape2.beta),
                              domain = beta.limits)
      reg.beta <- function(t) (t-a.beta)/(b.beta-a.beta)

      plot(trans.beta(Haz)~t, col=colour[ord%%n.col+1],
           main="beta", ylab="")
      lines(t,reg.beta(t))
    }, error = function(e) e)
  }
}
```

```
# Exponentiated Weibull
if(distributions[i]=="expweibull"){
  ord<-ord+1
  dexpwei <<- function(x,shape1,shape2,scale){
              shape2*shape1*scale^shape1*
               x^(shape1-1)*exp(-(scale*x)^shape1)*
               (1-exp(-(scale*x)^shape1))^(shape2-1)}
  pexpwei <<- function(q,shape1,shape2,scale){
                       (1-exp(-(scale*q)^shape1))^shape2}
  if(min(data[,1])<0){
    plot(1,1,xlim=c(0,1),ylim=c(0,1),col="grey100",
         main="exp-weibull", xlab="", ylab="")
    text(0.5,0.57, lab='The data is')
    text(0.5,0.43,lab='out of range')
  }
  else{
    tryCatch({
      fit.expwei <- fitdistcens(data,"expwei",
                                start=list(shape1=1,shape2=1,scale=1))
      shape1.expwei <- fit.expwei$estimate[1]
      shape2.expwei <- fit.expwei$estimate[2]
      scale.expwei <- fit.expwei$estimate[3]
      trans.expwei <- function(Haz){
                      log(-log(1-(1-exp(-Haz))^(1/shape2.expwei)))}
      parameters$expweibull <- c(shape1.expwei, shape2.expwei,
                                 scale.expwei)
      reg.expwei <- function(t) shape1.expwei*log(scale.expwei*t)

      plot(trans.expwei(Haz)~log(t),col=colour[ord%%n.col+1],
           main="exp-weibull", ylab="")
      lines(log(t),reg.expwei(t))
    }, error = function(e) e)
  }
}

# Exponential power
if(distributions[i]=="exppower"){
  ord<-ord+1
  dexppow <<- function(x,shape,scale){
    shape*scale^shape*x^(shape-1)*exp((scale*x)^shape)*
      exp(1-exp((scale*x)^shape))}
  pexppow <<- function(q,shape,scale) 1-exp(1-exp((scale*q)^shape))
  if(min(data[,1])<0){
    plot(1,1,xlim=c(0,1),ylim=c(0,1),col="grey100",
         main="exp-power", xlab="", ylab="")
    text(0.5,0.57, lab='The data is')
    text(0.5,0.43,lab='out of range')
  }
```

```
      else{
        tryCatch({
          fit.exppow <- fitdistcens(data,"exppow",
                                    start=list(shape=0.5,scale=0.5))
          shape.exppow <- fit.exppow$estimate[1]
          scale.exppow <- fit.exppow$estimate[2]
          trans.exppow <- function(Haz) log(log(Haz+1))
          parameters$exppower <- c(shape.exppow, scale.exppow)
          reg.exppow <- function(t) shape.exppow*log(scale.exppow*t)

          plot(trans.exppow(Haz)~log(t),col=colour[ord%%n.col+1],
               main="exp-power", ylab="")
          lines(log(t),reg.exppow(t))
        }, error = function(e) e)
      }
    }
  }
  options(digits=7)
  parameters
}
```

Appendix C
Grane.test code

```
Grane.test <- function(time, cens, distr, cens.type, cens.time,
                           beta.limits=c(0,1), Q.plot = "TRUE",
                           parameters = list(shape = NULL, shape2 = NULL,
                                             location = NULL, scale = NULL)
){
  # Load the required packages
  require(numDeriv)
  require(fitdistrplus)

  # Compute the sample and the uncensored observations sizes
  n <- length(time)
  r <- sum(cens)

  d<-data.frame(time=time, cens=cens)
  data <- data.frame(left=time,right=ifelse(cens==1,time,NA))

  if(cens.type == "I") y <- c(sort(d$time[d$cens==1]),cens.time)
  if(cens.type == "II") y <- sort(d$time[d$cens==1])

  alpha <- parameters$shape; gamma <- parameters$shape2
  mu <- parameters$location; beta <- parameters$scale

  # Compute x vector from the considered distribution

  if(distr=="weibull"){
    param<-fitdistcens(data,"weibull")
    if(is.null(alpha) || is.null(beta)){
      alpha<-unname(param$estimate[1])
      beta<-unname(param$estimate[2])
    }
    x <- pweibull(y,alpha,beta)
  }
```

```
  if(distr=="gumbel"){
    dgumbel <<- function(x,mu,beta){
                  1/beta*exp((x-mu)/beta)*exp(-exp((x-mu)/beta))}
    pgumbel <<- function(q,mu,beta) 1-exp(-exp((q-mu)/beta))
    param<-fitdistcens(data,"gumbel",start=list(mu=0,beta=2))
    if(is.null(mu) || is.null(beta)){
      mu<-unname(param$estimate[1])
      beta<-unname(param$estimate[2])
    }
    x <- pgumbel(y,mu,beta)
  }

  if(distr=="norm"){
    param<-fitdistcens(data,"norm")
    if(is.null(mu) || is.null(beta)){
      mu<-unname(param$estimate[1])
      beta<-unname(param$estimate[2])
    }
    x <- pnorm(y,mu,beta)
  }

  if(distr=="lnorm"){
    param<-fitdistcens(data,"lnorm")
    if(is.null(mu) || is.null(beta)){
      mu<-unname(param$estimate[1])
      beta<-unname(param$estimate[2])
    }
    x <- plnorm(y,mu,beta)
  }

  if(distr=="logis"){
    param<-fitdistcens(data,"logis")
    if(is.null(mu) || is.null(beta)){
      mu<-unname(param$estimate[1])
      beta<-unname(param$estimate[2])
    }
    x <- plogis(y,mu,beta)
  }

  if(distr=="loglogis"){
    dloglogis <<- function(x,alpha,beta){
                   alpha*beta^(-alpha)*x^(alpha-1)/(1+(x/beta)^alpha)^2}
    ploglogis <<- function(q,alpha,beta) 1/(1+(q/beta)^(-alpha))
    param<-fitdistcens(data,"loglogis",start=list(alpha=1,beta=1))
    if(is.null(alpha) || is.null(beta)){
      alpha<-unname(param$estimate[1])
      beta<-unname(param$estimate[2])
    }
```

```
  x <- ploglogis(y,alpha,beta)
}

if(distr=="beta"){
  a.beta<-beta.limits[1]
  b.beta<-beta.limits[2]
  param<-fitdistcens((data-a.beta)/(b.beta-a.beta),"beta")
  if(is.null(alpha) || is.null(gamma)){
    alpha<-unname(param$estimate[1])
    gamma<-unname(param$estimate[2])
  }
  x <- pbeta((y-a.beta)/(b.beta-a.beta), alpha, gamma)
}

if(distr=="expweibull"){
  dexpwei <<- function(x,alpha,gamma,beta){
                gamma*alpha*beta^alpha*x^(alpha-1)*
                  exp(-(beta*x)^alpha)*
                    (1-exp(-(beta*x)^alpha))^(gamma-1)}
  pexpwei <<- function(q,alpha,gamma,beta) (1-exp(-(beta*q)^alpha))^gamma
  param<-fitdistcens(data,"expwei",start=list(alpha=1,gamma=1,beta=1))
  if(is.null(alpha) || is.null(gamma) || is.null(beta)){
    alpha<-unname(param$estimate[1])
    gamma<-unname(param$estimate[2])
    beta<-unname(param$estimate[3])
  }
  x <- pexpwei(y, alpha, gamma, beta)
}

if(distr=="exppower"){
  dexppow <<- function(x,alpha,beta){
    alpha*beta^alpha*x^(alpha-1)*exp((beta*x)^alpha)*
      exp(1-exp((beta*x)^alpha))}
  pexppow <<- function(q,alpha,beta) 1-exp(1-exp((beta*q)^alpha))
  param<-fitdistcens(data,"exppow",start=list(alpha=0.5,beta=0.5))
  if(is.null(alpha) || is.null(beta)){
    alpha<-unname(param$estimate[1]); beta<-unname(param$estimate[2])
  }
  x <- pexppow(y, alpha, beta)
}

# Compute the Q statistic
r. <- length(x)
a <- rep(0,r.)
a[1:r.-1] <- 6*((2*(1:(r.-1))-1)*r.-n^2)/(n^2*r.)
a[r.] <- 6*(r.-1)*(n^2-r.*(r.-1))/(n^2*r.)

Q <- sum(a*x)
```

```
# pdf under H_0
b.values <- round(6/n^2*(2*(1:r.)-(1:r.)^2-1)+6/r.*((1:r.)-1),10)
nu <- as.numeric(table(b.values[b.values!=0]))
b <- as.numeric(dimnames(table(b.values[b.values!=0]))[[1]])
k<-length(b)

ind <- matrix(rep(1,k*k),ncol=k)-diag(k)
G.den <- function(s) apply(replace((s+1/b)^nu*ind,ind==0,1), 2, prod)

C.ctt<-1/prod(b^nu)

C<-matrix(rep(0,2*k), ncol=2)
for(j in 1:k){
  if(nu[j]==1){
    C[j,1]<-C.ctt*1/G.den(-1/b[j])[j]
  }
  if(nu[j]==2){
    C[j,]<-C.ctt*c(-jacobian(G.den,-1/b[j])[j]/(G.den(-1/b[j])[j])^2,
                   1/G.den(-1/b[j])[j])
  }
}

int.vec <- matrix(rep(0,2*k), ncol=2)
int.vec[,1] <- ifelse(b>0,0,b)
int.vec[,2] <- ifelse(b>0,b,0)
int <- c(min(int.vec[,1]),max(int.vec[,2]))


f_Q <- function(s){
  value <- array(rep(0,2*k*length(s)), dim=c(k,2,length(s)))
  for(l in 1:k){
    for(m in 1:2){
      if(C[l,m] != 0){
        value[l,m,] <- (sign(b[l])*C[l,m]*ifelse(s/b[l]>0 & s/b[l]<1,1,0)
                        *s^(m-1)*(1-s/b[l])^(n-m))/beta(m,n-m+1)
      }
    }
  }
  return(ifelse(n>14 & s<ifelse(n<17,0.45,0.5) & floor(n/2)+4<=r,
                ifelse(abs(apply(value,3,sum))<0.5 & n==r,
                       ifelse(n>=18 & s<0.35,0,apply(value,3,sum)),0),
                ifelse(apply(value,3,sum)<0 &
                       floor(n/2)+4<=r,0,apply(value,3,sum))))
}

options(digits=7)
Prob.Q <- integrate(f_Q, int[1], max(min(int[2],Q),0), subdivisions=2000)
```

```
  cat("\n Q statistic = ", Q)
  cat("\n P(x<Q) = ", Prob.Q$value, " with absolute error <",
       Prob.Q$abs.error,"\n\n")
  output <- list(test=c(Q.stat=Q,
                        p.value=Prob.Q$value,
                        abs.error=Prob.Q$abs.error),
                 param = c(shape = alpha, shape2 = gamma,
                           location = mu, scale = beta))
  if(Q.plot=="TRUE"){
    par(mfrow=c(1,1))
    points <- seq(int[1],int[2],0.0001)
    f.points <- f_Q(points)
    pointsQ <- seq(int[1],Q,0.0001)
    plot(points, f.points, type="l", xlab="Q", ylab="f(Q)")
    lines(pointsQ, f_Q(pointsQ), type="h", col="red")
    lines(points,f.points, type="l")
  }
  output
}
```

Appendix D
KScens code

```
KScens <- function(x, c, distr, beta.limits=c(0,1),
                   parameters = list(shape = NULL, shape2 = NULL,
                                     location = NULL, scale = NULL)
){
  # Load the required packages
  require(survival)
  require(fitdistrplus)

  n <- length(x)

  # Compute the decimal positions of the data
  dec<-max(apply(data.frame(x),1,
                 function(x)
                   nchar(toString(x))-nchar(toString(floor(x)))-1))
  if(dec>10){ n.dec <- dec+1}
  else{ n.dec <- 10}
  options(digits=n.dec)

  d<-data.frame(time=x, cens=c, count=rep(1,n))
  data<-data.frame(left=x,right=ifelse(c==1,x,NA))

  alpha <- parameters$shape; gamma <- parameters$shape2
  mu <- parameters$location; beta <- parameters$scale

  # Determine the theoretical distribution and estimate its parameters
  if(distr=="weibull"){
    f.surv <<- function(x, alpha, gamma, mu, beta)
                 1 - pweibull(x, alpha, beta)
    if(is.null(alpha) || is.null(beta)){
      param<-fitdistcens(data,"weibull")
      alpha<-unname(param$estimate[1])
      beta<-unname(param$estimate[2])
    }
  }
```

```
  if(distr=="gumbel"){
    dgumbel <<- function(x,mu,beta)
                  1/beta*exp((x-mu)/beta)*exp(-exp((x-mu)/beta))
    pgumbel <<- function(q,mu,beta) 1-exp(-exp((q-mu)/beta))
    f.surv <<-  function(x, alpha, gamma, mu, beta)
                  1 - pgumbel(x, mu, beta)
    if(is.null(mu) || is.null(beta)){
      param<-fitdistcens(data,"gumbel",start=list(mu=-3,beta=3))
      mu<-unname(param$estimate[1])
      beta<-unname(param$estimate[2])
    }
  }

  if(distr=="norm"){
    f.surv <<-  function(x, alpha, gamma, mu, beta)
                  1 - pnorm(x, mu, beta)
    if(is.null(mu) || is.null(beta)){
      param<-fitdistcens(data,"norm")
      mu<-unname(param$estimate[1])
      beta<-unname(param$estimate[2])
    }
  }

  if(distr=="lnorm"){
    f.surv <<-  function(x, alpha, gamma, mu, beta)
                  1 - plnorm(x, mu, beta)
    if(is.null(mu) || is.null(beta)){
      param<-fitdistcens(data,"lnorm")
      mu<-unname(param$estimate[1])
      beta<-unname(param$estimate[2])
    }
  }

  if(distr=="logis"){
    f.surv <<-  function(x, alpha, gamma, mu, beta)
                  1 - plogis(x, mu, beta)
    if(is.null(mu) || is.null(beta)){
      param<-fitdistcens(data,"logis")
      mu<-unname(param$estimate[1])
      beta<-unname(param$estimate[2])
    }
  }

  if(distr=="loglogis"){
    dloglogis <<- function(x,alpha,beta) {
                    alpha*beta^(-alpha)*x^(alpha-1)/(1+(x/beta)^alpha)^2}
    ploglogis <<- function(q,alpha,beta) 1/(1+(q/beta)^(-alpha))
```

```
  f.surv <<-  function(x, alpha, gamma, mu, beta)
                1 - ploglogis(x, alpha, beta)
  if(is.null(alpha) || is.null(beta)){
    param<-fitdistcens(data,"loglogis",start=list(alpha=1,beta=1))
    alpha<-unname(param$estimate[1])
    beta<-unname(param$estimate[2])
  }
}
if(distr=="beta"){
  a.beta<-beta.limits[1]
  b.beta<-beta.limits[2]
  f.surv <<-  function(x, alpha, gamma, mu, beta)
                1 - pbeta((x-a.beta)/(b.beta-a.beta), alpha, gamma)
  if(is.null(alpha) || is.null(gamma)){
    param<-fitdistcens((data-a.beta)/(b.beta-a.beta),"beta")
    alpha<-unname(param$estimate[1])
    gamma<-unname(param$estimate[2])
  }
}

if(distr=="expweibull"){
  dexpwei <<- function(x,alpha,gamma,beta){
                gamma*alpha*beta^alpha*x^(alpha-1)*exp(-(beta*x)^alpha)*
                  (1-exp(-(beta*x)^alpha))^(gamma-1)}
  pexpwei <<- function(q,alpha,gamma,beta) (1-exp(-(beta*q)^alpha))^gamma
  f.surv <<-  function(x, alpha, gamma, mu, beta)
                1 - pexpwei(x, alpha, gamma, beta)
  if(is.null(alpha) || is.null(gamma) || is.null(beta)){
    param<-fitdistcens(data,"expwei",start=list(alpha=1,gamma=1,beta=1))
    alpha<-unname(param$estimate[1])
    gamma<-unname(param$estimate[2]); beta<-unname(param$estimate[3])
  }
}

if(distr=="exppower"){
  dexppow <<- function(x,alpha,beta){
                alpha*beta^alpha*x^(alpha-1)*exp((beta*x)^alpha)*
                   exp(1-exp((beta*x)^alpha))}
  pexppow <<- function(q,alpha,beta) 1-exp(1-exp((beta*q)^alpha))
  f.surv <<-  function(x, alpha, gamma, mu, beta)
                1 - pexppow(x, alpha, beta)
  if(is.null(alpha) || is.null(beta)){
    param<-fitdistcens(data,"exppow",start=list(alpha=0.5,beta=0.5))
    alpha<-unname(param$estimate[1])
    beta<-unname(param$estimate[2])
  }
}
```

```
# Break the ties between censored and uncensored observations.
# Force that censored observations occur infinitesimally
# later that uncensored ones
time1<-sort(x)[1:n-1]
time2<-sort(x)[2:n]
diff.min <- min(abs(time1-time2)[abs(time1-time2)!=0])

aggr <- aggregate(. ~ time, data=d, FUN=sum)
pos<- which(d$time %in% aggr$time[aggr$count>1 &
                                  aggr$cens<aggr$count &
                                  aggr$cens>0])
d$time[pos] <- d$time[pos]+min(1/10^(n.dec-1),diff.min/2)*(1-d$cens[pos])


# Estimate the survival of the observed times
sum.survT<-summary(survfit(Surv(d$time, d$cens)~1, type='fh2'),
                   times=sort(d$time), extend=T)
survT<-unique(data.frame(time=round(sum.survT$time,dec),
              surv=sum.survT$surv))

# Estimate the survival of the censored times
sum.survC<-summary(survfit(Surv(d$time, 1-d$cens)~1, type='fh2'),
                   times=sort(d$time), extend=T)
survC<-unique(aggregate(. ~ time,
                        data = data.frame(time=round(sum.survC$time,dec),
                                          surv=sum.survC$surv),
                        FUN = min))
# distinct times vector
t.<-survT$time
m<-length(t.)

# Auxiliar vectors
t.ant<-c(0,survT$time[1:dim(survT)[1]-1])
survT.ant <- c(1,survT$surv[1:dim(survT)[1]-1])
survC.ant <- c(1,survC$surv[1:dim(survC)[1]-1])

# Compute A and B sumands
A.vec<-sqrt(survC.ant)*log(f.surv(t.ant,alpha,gamma,mu,beta)/
                           f.surv(t.,alpha,gamma,mu,beta))
A.vec[is.nan(A.vec)]<-0

B.vec<-sqrt(survC.ant)*log(survT.ant/survT$surv)
B.vec[is.nan(B.vec)]<-0
B.vec.ant<-c(0,B.vec[1:length(B.vec)-1])

# Compute the left- and the right-hand limit of Y at t_i
# (only for observed times)
```

```
  Y.left<-1/2*sqrt(n)*(survT.ant+f.surv(t.,alpha,gamma,mu,beta))*
          (cumsum(A.vec)-cumsum(B.vec.ant))*ifelse(B.vec>0,1,0)
  Y<-1/2*sqrt(n)*(survT$surv+f.surv(t.,alpha,gamma,mu,beta))*
     (cumsum(A.vec)-cumsum(B.vec))*ifelse(B.vec>0,1,0)

  # Compute also the value of Y at the last time t_m
  Ym<-1/2*sqrt(n)*(survT$surv[m]+f.surv(t.[m],alpha,gamma,mu,beta))*
      (cumsum(A.vec)[m]-cumsum(B.vec)[m])

  #Compute A and R
  A<-max(abs(c(Y.left,Y,Ym)))
  R<-1-1/2*(survT$surv[m]+f.surv(t.[m],alpha,gamma,mu,beta))

  options(digits=7)
  p <- function(y,x)
    1-pnorm(y/sqrt(x-x^2))+pnorm(y*(2*x-1)/sqrt(x-x^2))*exp(-2*y^2)
  if(p(A,R) <= 0.40){
    p.value <- 2*p(A,R)
    cat("\n p-value: ", p.value,'\n')
  }
  else{
    cat('\n Warning! The p-value can not be estimated
         because p(A,F(ym))>0.4 \n')
    p.value <- NULL
  }
  output <- list(test = c(p.value = p.value, "A" = A,
                          "F(ym)" = R, "ym" = t.[m]),
                 distr = distr,
                 param = c(shape = alpha, shape2 = gamma,
                           location = mu, scale = beta))
  output
}
```